

**ALGORITMO DE ESTIMACIÓN DE PARÁMETROS Y
MODULACIÓN DE UNA SEÑAL RECIBIDA POR UN SDR**

VERÓNICA TORO BETANCUR

**UNIVERSIDAD EAFIT
ESCUELA DE CIENCIAS
MAESTRÍA EN FÍSICA APLICADA
MEDELLÍN
2017**

VERÓNICA TORO BETANCUR
ALGORITMO DE ESTIMACIÓN DE PARÁMETROS Y
MODULACIÓN DE UNA SEÑAL RECIBIDA POR UN SDR

Proyecto de grado para optar al título de Magíster en Física Aplicada

Asesor: PhD José Ignacio Marulanda Bernal

UNIVERSIDAD EAFIT
ESCUELA DE CIENCIAS
MAESTRÍA EN FÍSICA APLICADA
MEDELLÍN
2017

“La ciencia es un esfuerzo de colaboración. Los resultados combinados de varias personas que trabajan juntas es a menudo mucho más eficaz de lo que podría ser el de un científico que trabaja solo.”

John Bardeen

AGRADECIMIENTOS

Existen muchas personas a las que debo agradecer no solo en el desarrollo de este trabajo de grado, sino en todo mi proceso académico y en mi vida personal.

En primer lugar agradezco al profesor José Ignacio Marulanda Bernal por el acompañamiento que me brindó en este trabajo de grado y por haber creído en mí y darme la oportunidad de desarrollar este proyecto y mis estudios en maestría. Gracias infinitas al estudiante del doctorado en ingeniería Augusto Carmona Valencia por estar siempre atento a mis inquietudes, por prestarme una ayuda invaluable al asesorarme con sus vastos conocimientos y por trabajar hombro a hombro conmigo.

Gracias a la Universidad EAFIT, específicamente al Departamento de Compras por estar siempre atentos a los requerimientos de este proyecto y al Departamento de Investigación por darme la beca con que llevé a cabo mis estudios.

A mis padres que siempre estuvieron incondicionalmente a mi lado y que se preocuparon para que me dedicara únicamente a mis estudios. A mi madre especialmente le agradezco que madrugara conmigo y se acostara tarde acompañándome y sin negarse nunca a prestarme su ayuda. A mi padre que toda su vida ha trabajado y lo sigue haciendo a pesar de sentirse cansado.

Agradecimientos también para el hombre que me prestó su vida, que me prestó su mano cuando no tenía ganas de seguir y que me levantó obligándome a continuar porque confiaba en mí. Que me brindó apoyo moral durante todo este proceso, que me enseñó a ser una mejor persona. Mi compañero incondicional, Marco.

CONTENIDO

1	INTRODUCCIÓN.....	11
1.1	PLANTEAMIENTO DEL PROBLEMA	11
1.2	JUSTIFICACIÓN	12
1.3	OBJETIVO GENERAL.....	12
1.4	OBJETIVOS ESPECÍFICOS	13
2	MARCO TEÓRICO	14
2.1	MODULACIÓN BFSK	14
2.2	MODULACIÓN BPSK	15
2.3	PROBLEMAS DE CLASIFICACIÓN.....	15
2.3.1	Redes neuronales	16
2.3.2	Máquinas de soporte vectorial.....	18
2.3.3	Árboles de decisión	23
3	ESTADO DEL ARTE	25
3.1	ESTIMACIÓN DE PARÁMETROS.....	25
3.2	CLASIFICACIÓN DE MODULACIÓN	28
4	ESTACIÓN DE TIERRA.....	30
4.1	CONFIGURACIÓN EN RECEPCIÓN.....	30
4.2	CONFIGURACIÓN EN TRANSMISIÓN.....	31
4.3	MODELO DE PÉRDIDAS DE LA ESTACIÓN DE TIERRA	33
4.4	CONTROL DE LA POSICIÓN DE LAS ANTENAS	34
4.5	MODELACIÓN DE LA ESTACIÓN DE TIERRA	37
5	METODOLOGÍA.....	40
5.1	CARACTERIZACIÓN DE LA POTENCIA DE RECEPCIÓN DEL RADIO	40
5.1.1	Montaje	40
5.1.2	Toma de datos.....	41
5.1.3	Resultados.....	42

5.2	CREACIÓN DE LOS SETS DE DATOS	44
5.2.1	BPSK	44
5.2.2	BFSK	46
5.2.3	Señales capturadas con el SDR	48
5.3	ALGORITMO DE ESTIMACIÓN DE PARÁMETROS	50
5.3.1	Amplitud.....	51
5.3.2	Frecuencia central.....	52
5.3.3	Ancho de banda	53
5.4	CLASIFICACIÓN DE MODULACIÓN	55
6	RESULTADOS	60
6.1	ALGORITMO DE ESTIMACIÓN DE PARÁMETROS	60
6.2	CLASIFICACIÓN DE MODULACIÓN	65
7	CONCLUSIONES.....	67
8	REFERENCIAS	69

LISTA DE FIGURAS

Figura 1. Representación de un modulador BPSK.....	15
Figura 2. Diagrama de redes neuronales.	17
Figura 3. Problema de clasificación de SVM.	19
Figura 4. Vectores de soporte.	20
Figura 5. Esquema de la estación de tierra.	32
Figura 6. Geometría de la ubicación de un satélite respecto a una estación de tierra.	35
Figura 7. Geometría de la posición de un satélite geoestacionario respecto a una estación de tierra.	36
Figura 8. Vista isométrica de la antena VHF.....	37
Figura 9. Detalle de la antena VHF.	37
Figura 10. Detalle de la sujeción al mástil de la antena VHF.	38
Figura 11. Detalle de las sujeciones de los elementos en la antena VHF.....	38
Figura 12. Vista isométrica de la antena UHF.....	38
Figura 13. Detalle de la antena UHF.	39
Figura 14. Detalle de la sujeción al mástil de la antena UHF.	39
Figura 15. Montaje para la caracterización de la potencia de recepción.	40
Figura 16. Datos de la caracterización y aproximación por regresión lineal en 145.5 MHz ..	42
Figura 17. Datos de la caracterización y aproximación por regresión lineal en 435 MHz ..	42
Figura 18. Datos de la caracterización y aproximación por regresión lineal en 2.46 GHz ..	43
Figura 19. Relación entre la amplitud marcada por el radio y la potencia de la señal, a diferentes frecuencias.	43
Figura 20. Señal BPSK generada con SNR de -20 dB.	45
Figura 21. Señal BPSK generada con SNR de 0 dB.	46
Figura 22. Señal BPSK generada con SNR de 18 dB.	46
Figura 23. Señal BFSK generada con SNR de -20 dB.	47
Figura 24. Señal BFSK generada con SNR de 0 dB.	48
Figura 25. Señal BFSK generada con SNR de 18 dB.	48
Figura 26. Señal real en VHF con modulación LoRa.	49
Figura 27. Señal real en UHF tomada con un offset de 25000 Hz y con modulación FSK.	49
Figura 28. Señal real en Banda S con modulación FSK.....	50
Figura 29. Esquema del algoritmo de estimación de parámetros.	55
Figura 30. Resultado del algoritmo de estimación de parámetros para una señal BPSK con SNR de 4 dB.	62
Figura 31. Resultado del algoritmo de estimación de parámetros para una señal BPSK con SNR de 18 dB.	63

Figura 32. Resultado del algoritmo de estimación de parámetros para una señal BFSK con SNR de -6 dB.....	63
Figura 33. Resultado del algoritmo de estimación de parámetros para una señal BFSK con SNR de 10 dB.	64
Figura 34. Resultado del algoritmo de estimación de parámetros para una señal BFSK con SNR de 18 dB.	64
Figura 35. Resultado del algoritmo de estimación de parámetros para una señal con modulación BFSK adquirida con un SDR.....	65

LISTA DE TABLAS

Tabla 1. Resultados de algoritmo NSD comparados con Detección de Energía.....	28
Tabla 2. Ganancias de los diferentes elementos de la estación de tierra.	34
Tabla 3. Parámetros del bloque de canal dinámico de GNU Radio.	45
Tabla 4. Conjuntos de características usados para entrenar los algoritmos de clasificación de modulación.	56
Tabla 5. Porcentaje en los que el algoritmo de estimación de parámetros converge a una solución para señales BPSK a determinado SNR y los porcentajes de error absoluto para cada parámetro.	60
Tabla 6 Porcentaje en los que el algoritmo de estimación de parámetros converge a una solución para señales BFSK a determinado SNR y los porcentajes de error absoluto para cada parámetro.	61
Tabla 7. Resultados del algoritmo de estimación de parámetros para datos adquiridos por el SDR.	62
Tabla 8. Mejor resultado para el algoritmo de Redes Neuronales.	65
Tabla 9. Mejor resultado para el algoritmo de Árboles de Decisión.	66
Tabla 10. Mejor resultado para el algoritmo de Máquinas de Soporte Vectorial.....	66

RESUMEN

En el presente trabajo se plantea desarrollar un algoritmo capaz de determinar la amplitud, la frecuencia central, el ancho de banda y la modulación de una señal recibida por un SDR (*Software Defined Radio*) en donde las modulaciones trabajadas fueron BPSK (*Binary Phase-Shift Keying*) y BFSK (*Binary Frequency-Shift Keying*). Para esto se dividió el problema en dos sub-algoritmos: uno para la estimación de parámetros y otro para la clasificación de la modulación de la señal.

Para la estimación de parámetros se utilizaron diferentes modelos para aproximar cada uno de los estos: amplitud, frecuencia y ancho de banda. Luego de crear los modelos, se usaron de forma iterativa hasta satisfacer una condición de error, después de lo cual, el algoritmo entrega los parámetros. Éste fue probado a diferentes valores de SNR (*signal-to-noise ratio*) usando, tanto señales generadas desde el computador, simulando diferentes condiciones de ruido y de canal de transmisión, como con señales adquiridas desde un SDR.

Con el fin de clasificar la modulación de las señales en BPSK o BFSK se utilizaron tres algoritmos de *machine learning*: redes neuronales, máquinas de soporte vectorial y árboles de decisión. Con cada uno de estos algoritmos se evaluaron diferentes formas de representar las modulaciones, con ayuda de funciones matemáticas comunes. Finalmente se presentan los porcentajes de aciertos obtenidos con cada algoritmo y característica de los datos, para concluir que la mejor solución a este problema es el modelo de redes neuronales con una característica lineal de la desviación estándar de la fase instantánea de la señal evaluada.

CAPÍTULO 1

INTRODUCCIÓN

En el presente trabajo de grado se muestran los desarrollos realizados durante el transcurso de los estudios de maestría. Primero se presenta el problema, definiendo bien sus características y los objetivos que se plantearon en el inicio del proyecto de grado. Luego, en el Marco Teórico, se hace un recorrido por algunos términos y conceptos necesarios para entender el problema atacado. En el Estado del Arte se detallan algunos desarrollos similares resueltos con anterioridad y que se relacionan con el presente proyecto, dando también brevemente un reporte de los resultados obtenidos por los trabajos que allí se mencionan.

En el capítulo llamado Estación de Tierra se muestra el detalle de la estación implementada también durante los estudios de maestría y desde la cual se plantea el desarrollo del proyecto expuesto aquí.

Se entrará en materia cuando en el capítulo de Metodología se presenten los métodos llevados y cabo en las diferentes etapas del proyecto para que, con esto en mente, se pase a los Resultados en donde de forma concisa se muestra lo que se obtuvo con los desarrollos hechos. Finalmente, en el capítulo de Conclusiones, se hace un recorrido sobre los aprendizajes obtenidos con la implementación de este trabajo de grado.

1.1 PLANTEAMIENTO DEL PROBLEMA

Se propone realizar un algoritmo que permita procesar las señales recibidas por un radio SDR (*Software Defined Radio*), las cuales pueden provenir de diferentes elementos como drones, CubeSats, globos, cometas o cohetes implementados; y con dos tipos diferentes de modulaciones: BFSK (*Binary Frequency-Shift Keying*) y BPSK (*Binary Phase-Shift Keying*).

Con el procesamiento de las mismas se busca determinar los parámetros de amplitud, frecuencia central, ancho de banda y tipo de modulación. La identificación de la modulación se hará por medio de tres algoritmos de *Machine Learning*: Máquinas de Soporte Vectorial,

Árboles de Decisión y Redes Neuronales para finalmente tomar una decisión acerca de cuál de los algoritmos conlleva a un número de aciertos mayor para las modulaciones usadas.

La identificación de los diferentes parámetros de una señal recibida permite tener información suficiente para la traducción de los datos enviados. Esto es de gran utilidad en una situación en la que no se conocen las fuentes de las señales recibidas por el radio, como por ejemplo, cuando se trabaja en un ancho de banda libre que está por lo general muy saturado, es necesario reconocer cuál es la señal que contiene la información que es de interés para el usuario.

1.2 JUSTIFICACIÓN

El proyecto planteado encuentra aplicación en la estación de tierra que se está instrumentado en la Universidad EAFIT para comunicación bidireccional con dispositivos de observación de la tierra tales como drones, vehículos aéreos no tripulados (UAVs), cometas y globos instrumentados, satélites, entre otros. El desarrollo planteado en este informe, permitiría crear, en dicha estación, la capacidad de reconocimiento de los diferentes dispositivos que se conectarían a la misma, así como la determinación de parámetros esenciales para el establecimiento de comunicación, como lo son la amplitud de la señal, su frecuencia central, ancho de banda y modulación.

Resulta de gran relevancia que un desarrollo como éste se encuentre presente en una estación de tierra, ya que esto le daría versatilidad a la misma, haciendo que no sea necesario entregarle todos los parámetros del dispositivo emisor y que, por lo tanto, pueda comunicarse o detectar señales provenientes de emisores de los cuales no se conocen todas sus características, como satélites, por ejemplo. Esto encuentra aplicación también en labores de caracterización de espectro, en donde se quiere conocer cuántos dispositivos están emitiendo en una determinada banda y, de esta forma, conocer si existen señales no permitidas.

Se plantea el reconocimiento de dos diferentes tipos de modulación digital: BFSK y BPSK dado que estas son las más comunes en las comunicaciones amateur, por lo cual, con la terminación del proyecto se tendría la capacidad de establecer comunicación bidireccional con un gran número de dispositivos y, además, podría ser luego extendido a más tipos de modulaciones.

1.3 OBJETIVO GENERAL

Determinar la amplitud, frecuencia, ancho de banda y modulación de una señal recibida por un radio SDR.

1.4 OBJETIVOS ESPECÍFICOS

- Determinar los parámetros de amplitud, frecuencia central y ancho de banda de una señal recibida por un radio SDR, por medio de métodos iterativos de estimación.
- Clasificar entre las modulaciones BFSK y BPSK de una señal recibida por un radio SDR, usando los algoritmos de identificación de Máquinas de Soporte Vectorial, Árboles de Decisión y Redes Neuronales
- Determinar cuál de los algoritmos de identificación de modulación usados tiene mayor número de aciertos para las modulaciones usadas.

CAPÍTULO 2

MARCO TEÓRICO

2.1 MODULACIÓN BFSK

BFSK (*Binary Frequency-Shift Keying*) es una modulación digital en frecuencia, es decir que cada bit enviado es representado con una frecuencia diferente en la señal transmitida. Al ser una modulación binaria, cada símbolo es un bit, esto es, los bits son enviados uno a uno con una forma de onda que se puede representar como en las Ecuaciones (1) y (2).

$$s_1(t) = A \cos(2\pi f_1 t + \Phi_1), \quad (1)$$

$$s_2(t) = A \cos(2\pi f_2 t + \Phi_2). \quad (2)$$

Donde f_1 y f_2 son las frecuencias para el bit 1 y 0 respectivamente. Si Φ_1 y Φ_2 en $t = 0$, no son iguales, se tiene una modulación FSK no coherente lo cual ocurre cuando las señales s_1 y s_2 se generan en fuentes diferentes [1].

Para el caso de una modulación coherente, es decir que las señales s_1 y s_2 tienen la misma fase inicial, se podrían tener también frecuencias ortogonales, que son aquellas con las que se cumple la relación de la Ecuación (3).

$$\int_{kT}^{(k+1)T} s_1(t)s_2(t)dt = 0. \quad (3)$$

Se puede demostrar que para el caso de modulación coherente se tiene que cumplir que [2]

$$f_1 - f_2 = \frac{m}{2T}, \quad m \in \mathbb{Z}, \quad (4)$$

es decir que la separación entre las dos frecuencias debe ser un múltiplo entero de $1/2T$, donde T es el periodo de un bit.

2.2 MODULACIÓN BPSK

BPSK (*Binary Phase-Shift Keying*) es una modulación digital en donde se usa la fase de una señal para representar los bits 1 y 0. Dado que en este caso solo es necesario transmitir dos símbolos (bits) la separación entre fases de máxima (180°) y por tanto las probabilidades de error en la demodulación son mínimas [3]. Las señales para cada caso de fase, se pueden expresar como en las Ecuaciones (5) y (6), donde s_1 representa la señal para el bit 0 y s_2 la señal para el bit 1.

$$s_1(t) = A \cos(2\pi f_c t), \quad (5)$$

$$s_2(t) = A \cos(2\pi f_c t + \pi) = -A \cos(2\pi f_c t). \quad (6)$$

donde A es la amplitud de la señal, f_c es la frecuencia de la portadora. Es importante resaltar que estos dos parámetros son siempre constantes para el caso de modulación BPSK. Estas señales también podrían ser escritas como en la Ecuación (7)

$$s_1(t) = g(t) \cos(2\pi f_c t), \quad (7)$$

donde $g(t) = \pm A$ y de esta manera es claro representar esta modulación con un generador de señal NRZ (*non-return to zero*), en donde a los bits 0 les corresponde un pulso negativo y a los bits 1, un pulso positivo [4]. Esta representación puede verse en la Figura 1.

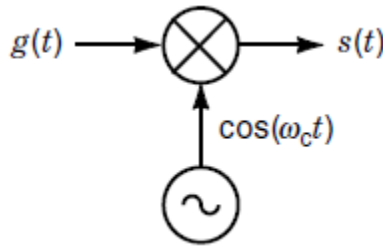


Figura 1. Representación de un modulador BPSK [4].

2.3 PROBLEMAS DE CLASIFICACIÓN

En los tipos de algoritmos de *Machine Learning*, se encuentra una categoría llamada algoritmos de clasificación, los cuales son destinados a aprender un mapeo entre unas entradas x y unas salidas y las cuales son del tipo $y \in \{1, \dots, C\}$ donde C es el número de clases en las que se quieren etiquetar los datos de entrada. En el caso en que $C = 2$ se tiene un problema de clasificación binario, en donde y se reduce a $y \in \{0, 1\}$. Si $C > 2$ se tiene un

problema de clasificación multiclase y, si las clases no son excluyentes entre sí, es decir, si varias clases se pueden cumplir para un mismo set de datos, entonces se está ante un problema de clasificación de múltiple etiquetas [5].

En el problema que se quiere abordar, solo se desarrollarán algoritmos de clasificación binarios ya que solo se tienen dos modulaciones como etiquetas. Los algoritmos que se plantea usar para este objetivo se describen brevemente en las siguientes sub-secciones.

2.3.1 Redes neuronales

Una red neuronal artificial se compone de una capa de entradas, la cual tiene tantos elementos, como dimensiones tienen los datos de entrada; una o varias capas ocultas de neuronas que también se conocen como perceptrones y una capa de salida que contiene tantos elementos como clases se tengan en el problema de clasificación. Todos estos elementos se conectan entre sí por medio de unos pesos y unos *bias*, que son los valores que deben ser calculados en la etapa de entrenamiento y que se usan luego en el modelo que se utiliza en la etapa de predicción de la modulación.

Los problemas de clasificación binarios en las redes neuronales se reducen a determinar los pesos y los *bias* que correctamente predicen la clase de los datos de entrada. Para este propósito se utilizan algoritmos como *forward propagation* y *backpropagation*, siendo este último un método de aprendizaje supervisado, en donde, a partir de los datos obtenidos por la red neuronal y los datos reales que debieron haber sido obtenidos, se calcula un error, el cual se propaga hacia atrás y, proporcionalmente a este, se van corrigiendo los valores hasta llegar a un error final pequeño o aceptable [6].

Una vez se llega a los niveles de error aceptables, se guardan los valores de pesos y *bias* en el modelo, el cual luego se encargará de tomar datos nuevos y clasificarlos según las probabilidades que estos obtengan al final para cada clase.

La Figura 2 muestra un esquema de redes neuronales para un problema de clasificación binario con datos de entrada en dos dimensiones, representados por X y Y. Cada una de las líneas que conectan elementos entre sí, corresponden a un peso W y b_1, b_2, \dots, b_n son los *bias*, los cuales funcionan como un offset que opera en las entradas a las neuronas en la capa oculta y en las salidas. Cada neurona en las capas ocultas, es en realidad una función de activación que actúa sobre los datos de entrada a las mismas. Para estas funciones normalmente se utiliza $\tanh(x)$ o $\text{sigmoid}(x)$ ya que tienen una transición suave cuando se pasa de valores positivos a negativos, además de que tienen la propiedad de que su derivada está en términos de la misma función, por lo que se ahorran procesos de cálculo puesto que se calcula el valor de la función una vez y se reutiliza en el cálculo de la derivada. La derivada es importante porque ésta es usada en el método de aprendizaje supervisado de *backpropagation*, el cuál es ampliamente implementado. Finalmente, en las salidas, se utiliza

una función que traduzca los valores de entrada en probabilidades; para esto se suelen usar la función logística o la *softmax* que es una generalización de la anterior [7].

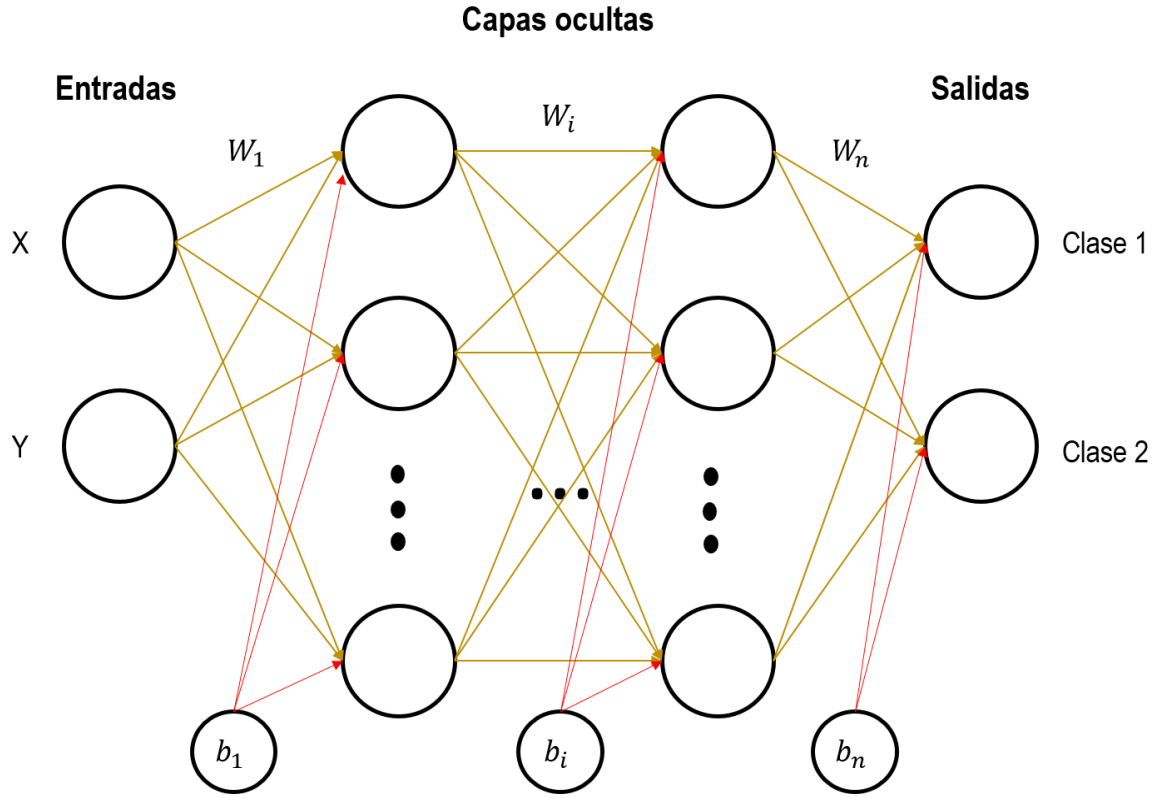


Figura 2. Diagrama de redes neuronales.

A continuación se muestra un planteamiento matemático de un problema de redes neuronales con k capas y l clases. Sea X una matriz con dimensiones $(n \times m)$ donde n es el número de datos de entrada y m las dimensiones de los mismos. Si $a_{i,1}$ es la función de activación de la i -ésima neurona de la primera capa, entonces

$$a_{i,1} = \tanh(z_1), \quad (8)$$

donde $z_1 = XW_1 + b_1$ es la entrada de la neurona $a_{i,1}$, W_1 es la matriz de pesos de la capa de entrada hacia la primera capa de neuronas y b_1 es el *bias* usado en dicha transición. Más generalmente se tiene

$$z_j = a_{j-1}W_j + b_j, \quad (9)$$

$$a_{i,j} = \tanh(z_j), \quad (10)$$

Para la última capa se cumple que

$$a_{k,1,\dots,l} = \text{softmax}(z_{k-1}), \quad (11)$$

Este planteamiento muestra la forma de recorrer la red neuronal desde el principio hasta el final, lo que se conoce como *forward propagation*. Ahora bien el método de recorrer la red desde las salidas hacia las entradas es el mencionado *backpropagation*. Para esto se hace un *forward propagation* inicial con los datos de entrenamiento y unos valores iniciales de pesos y *bias* para calcular en la salida el error entre el resultado obtenido y el deseado. Este error también se puede ver como el *log likelihood* negativo de la Ecuación 12.

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{n \in N} \sum_{i \in C} y_{n,i} \log(\hat{y}_{n,i}). \quad (12)$$

Esta ecuación anterior se puede ver como una pérdida, la cual incrementa solo cuando se tiene una mala predicción. La idea ahora es minimizar las pérdidas y para este problema de optimización se suelen usar diferentes variaciones de *gradient descent*, tales como *Stochastic Gradient Descent* (SGD) [8], *Minibatch Gradient Descent* [9] o simplemente *Gradient Descent* en donde el vector de derivadas se conforma con las derivadas de la función de pérdidas de la Ecuación (12) respecto a los diferentes pesos y *bias* que son los parámetros que se quieren encontrar.

$$\nabla L(y, \hat{y}) = \left(\frac{\partial L}{\partial W_1}, \frac{\partial L}{\partial W_2}, \dots, \frac{\partial L}{\partial W_k}, \frac{\partial L}{\partial b_1}, \frac{\partial L}{\partial b_2}, \dots, \frac{\partial L}{\partial b_k} \right). \quad (13)$$

Para encontrar estas derivadas se recurre a la regla de la cadena y así éstas quedan en términos de las funciones de activación cuyas derivadas son conocidas. El algoritmo de *backpropagation* se convierte en un proceso iterativo en donde se llega a optimizar la función de pérdidas y con ello se encuentran los parámetros W y b que se estaban buscando.

El número de iteraciones del algoritmo de redes neuronales en la etapa de entrenamiento es uno de los parámetros que deben ser ingresados por el usuario. Así como la tasa de aprendizaje del *gradient descent* que es un valor por el cual se multiplican los valores de W y b cada vez que se hace una nueva iteración; se puede ver como una medida del paso de las iteraciones. Estos parámetros determinan qué tan cercano se podría estar de la solución óptima para la función de pérdidas.

2.3.2 Máquinas de soporte vectorial

Las máquinas de soporte vectorial son un algoritmo de *Machine Learning* ampliamente usado para problemas de clasificación y regresión. En problemas de clasificación, éste debe ser

entrenado por métodos de aprendizaje supervisado ya que, de lo contrario, debería realizarse previamente una etapa de clusterización, con lo cual tomaría el nombre de *Support vector clustering* [10].

Con este algoritmo se busca encontrar un hiperplano que separe los datos k-dimensionales que pertenecen a diferentes clases y además se busca que este hiperplano sea el que mejor generalice la clasificación; esto se da cuando se maximiza la distancia entre los puntos que pertenecen a cada clase y el hiperplano creado.

Finalmente lo que se quiere obtener con este método para tomar decisiones acerca de los datos que se quieren clasificar es una expresión matemática. Para esto considere el caso de la Figura 3 en donde los datos graficados corresponden a dos clases diferentes (círculos verdes y cruces rojas) y son divididos por la línea azul.

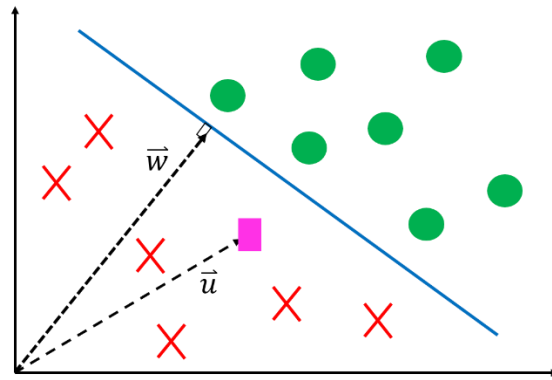


Figura 3. Problema de clasificación de SVM.

Si en la Figura 3 se quisiera determinar a qué clase pertenece el cuadro rosado lo que se hace es proyectar el vector \vec{u} que va desde el origen hasta la posición del cuadro, con el vector \vec{w} que va desde el origen e incide perpendicularmente en la frontera de decisión [11]. Esta proyección no es más que un producto punto entre ambos vectores y, si esta operación arroja un número mayor a cero, se diría que el cuadro rosado pertenece a la clase de los círculos verdes, mientras que si el resultado es negativo su clase sería la de las cruces rojas. Si el producto punto es igual a cero es porque el punto evaluado se encuentra en la frontera de decisión. Estos tres casos se expresan en las Ecuaciones (14), (15) y (16).

$$\vec{u} \cdot \vec{w} + b > 0, \quad \text{Clase -1} \quad (14)$$

$$\vec{u} \cdot \vec{w} + b < 0, \quad \text{Clase +1} \quad (15)$$

$$\vec{u} \cdot \vec{w} + b = 0, \quad \text{Frontera de decisión} \quad (16)$$

Se puede ver en las ecuaciones anteriores la presencia de una constante b la cual corresponde a un *bias* que, en el caso de un problema en dos dimensiones, se representaría como el intercepto en el eje y .

El problema de entrenamiento se reduce entonces a encontrar los parámetros \vec{w} y b del hiperplano para los cuales se da la clasificación y la separación entre los datos y el hiperplano es máxima. Para esto se llega a un problema de optimización con restricciones que surgen al considerar los vectores \vec{X}_+ y \vec{X}_- conocidos como los vectores de soporte, que son aquellos formados por los datos de entrenamiento que influyen en la formación del hiperplano, es decir, puede existir una gran cantidad de datos de entrenamiento pero solo algunos de ellos se encuentran cercanos a la frontera de decisión y, si estos datos se moviesen, la frontera de decisión cambiaría. Si \vec{X}_+ es el vector de soporte positivo, esto es, aquel que contiene las características de clase positiva y \vec{X}_- es el vector de soporte negativo que contiene las características con clase negativa, se pueden escribir las Ecuaciones (17) y (18).

$$\vec{X}_+ \cdot \vec{w} + b > 1, \quad (17)$$

$$\vec{X}_- \cdot \vec{w} + b < -1. \quad (18)$$

Esta situación es ilustrada en la Figura 4.

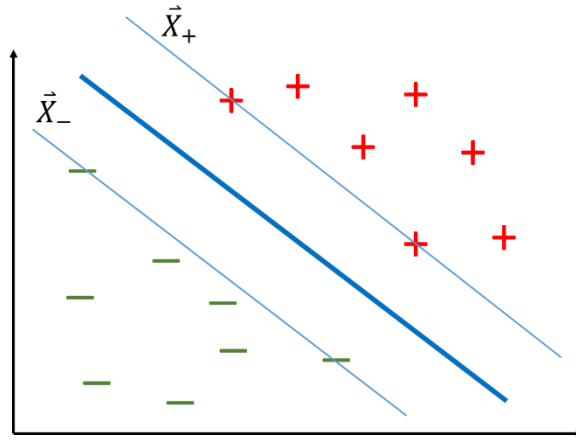


Figura 4. Vectores de soporte.

Ahora bien, si definimos y_i como la clase de los datos, es decir y_i puede ser 1 o -1 y multiplicamos las Ecuaciones (17) y (18) en ambos lados por y_i obtenemos los siguiente

$$y_i^+ (\vec{X}_+ \cdot \vec{w} + b) > y_i^+ (1) = 1, \quad (19)$$

$$y_i^- (\vec{X}_- \cdot \vec{w} + b) > y_i^- (-1) = 1, \quad (20)$$

donde $y_i^+ = 1$ y $y_i^- = -1$. En ambos casos las ecuaciones son iguales a 1, por lo que se obtiene, en la Ecuación (21) que es la restricción para el problema de optimización.

$$y_i (\vec{X}_i \cdot \vec{w} + b) - 1 \geq 0. \quad (21)$$

El problema de optimización se centra en encontrar un $\|\vec{w}\|$ mínimo ya que lo que se quiere es maximizar la distancia entre la frontera de decisión y los vectores de soporte y se tiene que la distancia entre los vectores de soporte es

$$d = (\vec{X}_+ - \vec{X}_-) \cdot \frac{\vec{w}}{\|\vec{w}\|} \quad (22)$$

ya que es la resta entre los vectores en la dirección del vector unitario $\vec{w}/\|\vec{w}\|$. Despejando \vec{X}_+ de la Ecuación (19), \vec{X}_- de la Ecuación (20) y reemplazando ambos valores en la Ecuación (22), se obtiene

$$d = \frac{2}{\|\vec{w}\|} \quad (23)$$

y como se quiere que d sea máxima, es por eso que se debe minimizar $\|\vec{w}\|$. Ahora bien, el problema de optimización será abordado con multiplicadores de Lagrange [12]. La Ecuación (24) muestra la expresión con la que se trabaja

$$\mathcal{L}(\vec{w}, b, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i (\vec{X}_i \cdot \vec{w} + b) - 1] \quad (24)$$

en donde solo se usa α como multiplicador de Lagrange dado que solo se tiene una restricción de inequidad. Se deriva la ecuación anterior respecto a \vec{w} y b y se igualan ambas expresiones a cero.

$$\frac{\partial \mathcal{L}(\vec{w}, b, \alpha)}{\partial \vec{w}} = \vec{w} - \sum_{i=1}^n \alpha_i y_i \vec{X}_i = 0 \quad (25)$$

Por lo que

$$\vec{w} = \sum_{i=1}^n \alpha_i y_i \vec{X}_i. \quad (26)$$

Mientras que para b se tiene

$$\frac{\partial \mathcal{L}(\vec{w}, b, \alpha)}{\partial b} = - \sum_{i=1}^n \alpha_i y_i = 0. \quad (27)$$

Reemplazando en (24) lo obtenido en (26) y (27) se obtiene

$$\mathcal{L}(\vec{w}, b, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j (\vec{X}_i \cdot \vec{X}_j). \quad (28)$$

Se puede ver que el problema de optimización que se tiene es un problema convexo dado que $\|\vec{w}\|$ es una función convexa [13] y se tienen restricciones tanto de inequidad como de igualdad. Por lo tanto, con miras a la programación del método de máquinas de soporte vectorial, el problema de entrenamiento se reduce a encontrar el punto más bajo de la función convexa $\|\vec{w}\|$ y esto se puede lograr al iterar con diferentes valores y diferentes tamaños de paso para asegurarse de encontrar el mínimo global. También se itera con diferentes valores de b hasta encontrar aquella pareja de \vec{w} y b que cumpla con la restricción (21) y en que además sea \vec{w} mínimo.

Luego de entrenar el algoritmo de máquinas de soporte vectorial, lo que sigue es la etapa de predicción con los *sets* de datos de prueba. Para determinar si unos datos de entrada \vec{x} hacen parte de la clase 1 o -1 basta con determinar el signo de la expresión $\vec{w} \cdot \vec{x} + b$, como lo muestra la Ecuación (29).

$$y = \text{sign}(\vec{w} \cdot \vec{x} + b). \quad (29)$$

Suele ocurrir que el problema que se quiere abordar no es linealmente separable, es decir, para unos datos dados en un espacio de dimensiones finitas, no existe un hiperplano que correctamente separe las clases del problema dado, para esto se pueden aumentar las dimensiones de los datos hasta un punto en donde estos sí sean linealmente separables. Esto se hace con el uso de unas funciones conocidas como *kernels* quienes se aplican sobre funciones cuyas operaciones son productos punto y como hemos visto, el problema de máquinas de soporte vectorial se basa en productos punto. Una función kernel k se puede definir como en la Ecuación (30).

$$k(x, x') = z \cdot z', \quad (30)$$

donde z y z' pertenecen al espacio de dimensiones al que se quieren llevar x y x' . Existen diferentes tipos de *kernels*: *Radial Basis Function* (RBF) [13], *Fisher kernel* [14], *Polynomial kernel* [15], *Smoother* [16], *String kernel* [17]. El *kernel* polinomial, por ejemplo, se define como sigue

$$k(x, x') = (1 + x_1 x'_1 + \dots + x_n x'_n)^p, \quad (31)$$

donde n es el número de dimensiones a que se quiere llegar y p es el grado del polinomio. Note que, sin embargo, todos los cálculos se realizan con x y x' . El *kernel* RBF se define como [18]

$$k(x, x') = e^{-\gamma \|x - x'\|^2}, \quad (32)$$

allí se puede ver que $\|x - x'\|^2$ es una distancia euclidiana y γ es un parámetro.

Otro concepto importante en las máquinas de soporte vectorial es el de *soft margin* y *hard margin*, estas se dan si se tiene una tolerancia en el entrenamiento o no, es decir, cuando se usan todos los datos de cada clase que se encuentran cercanos a la frontera de decisión como vectores de soporte, se da *hard margin* y esto conlleva a un sobreentrenamiento de los datos, sin embargo cuando se permite que algunos datos de una clase hagan parte de una clase diferente durante el proceso de entrenamiento se tiene *soft margin* y se maneja como una tolerancia ϵ en la restricción de la Ecuación (21) resultando en la Ecuación (33).

$$y_i(\vec{X}_i \cdot \vec{w} + b) \geq 1 - \epsilon. \quad (33)$$

2.3.3 Árboles de decisión

Los árboles de decisión son una herramienta predictiva que parte de un modelo en árbol en donde los nodos representan condiciones sobre las cuales se dividen los datos en ramas hasta llegar a los nodos terminales (aquellos desde los cuales no salen más ramas) en donde se determina la clase de los datos [19].

En aplicaciones de *Machine Learning*, se podría entrenar un árbol de decisión por medio de aprendizaje supervisado en donde, con un set de datos de entrenamiento, se compare cada uno de los valores de los datos en el set con alguno de los datos en el mismo, de esta manera se generan subgrupos de datos que van formando los nodos del árbol. La forma de determinar cuál será la característica con la cual se dividirán los datos es probar cada una de las posibles divisiones y escoger aquella que represente un menor costo. El costo de estas divisiones se suele definir por la entropía de los datos allí, es decir, observando qué tan mezcladas están las clases en cada subgrupo, entre menos mezcladas estén, menor entropía y por lo tanto mejor es el subgrupo [20].

De los posibles métodos de evaluación de costo, uno de los más usados es el coeficiente Gini que se muestra en la Ecuación (34) y que da una idea de la dispersión de las clases presentes en el subgrupo evaluado.

$$G = \sum_{i=0}^n p_i(1 - p_i). \quad (34)$$

Allí n es el número de clases y p_i es el porcentaje de presencia de la i -ésima clase en el grupo evaluado, es decir, p_i es la razón entre la cantidad de datos con clase i y el número total de datos en ese grupo de datos. Es claro que para cuando el grupo está compuesto por datos que tienen una misma clase ocurre que $G = 0$ puesto que $1 - p_i = 0$, mientras que en el peor de

los casos, en donde cada clase tienen la misma presencia en el grupo, se tiene que $G = 0.5$ ya que

$$G = 0.5(1 - 0.5) + 0.5(1 - 0.5) = 0.5. \quad (35)$$

En la Ecuación (35) se hizo el cálculo para un problema con dos clases. Se podría obtener otra función de costo al calcular la desviación estándar de las clases en cada subgrupo [21].

Ahora bien, se debe imponer un criterio de parada para el algoritmo de árboles de decisión para determinar qué tan grande puede ser y, con esto, qué tan preciso o sobreentrenado podría estar el árbol construido. Los criterios usados comúnmente son: el número de datos mínimo en los nodos terminales, el número máximo de niveles o capas que tendrá el árbol o el número de iteraciones del algoritmo. Se puede hacer también que el árbol sea construido de una forma eficiente para lo cual se puede recurrir a métodos de podado en donde se determine si es importante o no expandir más un nodo [22].

Una vez construido el árbol es hora de predecir las clases de nuevos datos, para esto los datos deben recorrer el árbol de acuerdo a las condiciones planteadas por cada nodo, es decir, si la característica de un nodo es la comparación numérica del dato con un valor, entonces éste hará parte del subgrupo de la derecha o la izquierda dependiendo de si es mayor o menor que el valor con el que se le compara. De esta manera recorre el árbol hasta llegar a los nodos terminales en donde se decide su clase.

El método de árboles de decisión es usado en aplicaciones como selección de variables: en donde muchos aspectos son tenidos en cuenta pero solo unos pocos son realmente útiles, los árboles de decisión pueden ayudar a determinar cuáles variables sí son importantes; clasificar valores que se salen de la media de los datos, para evitar que estos sean simplemente borrados, lo cual podría introducir ruido; y por supuesto la aplicación más popular: predecir estados futuros de una variable o sistema [23].

CAPÍTULO 3

ESTADO DEL ARTE

El estado del arte está dividido en las dos subsecciones en que se piensa que sea atacado el problema: estimación de parámetros de la señal transmitida y clasificación de la modulación de la misma. Desde ambos frentes se busca mostrar un panorama de los trabajos realizados en dichos temas y una manera de plantear una metodología a partir de los resultados reportados en la literatura.

3.1 ESTIMACIÓN DE PARÁMETROS

El problema de estimación de parámetros de una señal, también conocido como problema de sensado de espectro, es abordado principalmente en redes de radios cognitivos [24], [25] en donde se busca detectar señales presentes en el ancho de banda que usa la red para optimizar el uso de la misma, al emitir en las frecuencias menos usadas. Los radios cognitivos fueron propuestos inicialmente por Mitola en [26] y a partir de entonces este ha sido un tema muy trabajado y han surgido diferentes propuestas.

Los métodos más usados en aplicaciones de radios cognitivos, para la estimación de parámetros de una señal, son los métodos de detección de energía, métodos de filtro adaptado, detección de característica cicloestacionaria y detección basada en autovalores. Se encontró además el método iterativo con funciones de verosimilitud implementado por Curtis M. Watson en [27] en el cual propone la estimación de diversos parámetros de una señal recibida. Algunos trabajos relacionados a estos métodos serán descritos brevemente a continuación.

El método de detección de energía es el más usado en aplicaciones de sensado de espectro, los cuales, en general, pueden ser modelados como un problema binario de prueba de hipótesis, donde el conjunto de hipótesis manejadas se muestra en la Ecuación (36).

$$\begin{cases} H_0: y(t) = w(t) \\ H_1: y(t) = h \cdot x(t) + w(t) \end{cases} \quad (36)$$

Donde H_0 representa la hipótesis nula, es decir, cuando no hay señal transmitida, H_1 es la hipótesis para señal transmitida, $y(t)$ es la señal recibida, $x(t)$ es la señal transmitida, $w(t)$ es ruido gaussiano blanco aditivo (AWGN) con media cero y h es la ganancia del canal por el que fue transmitida la señal.

El método de detección de energía consiste en medir la energía promedio de la señal recibida por medio de un radiómetro, de manera que, en el momento en que haya presencia de una señal diferente a ruido, la energía aumenta y, si esta supera un valor de referencia *Threshold*, se determina que hay señales recibidas que deben ser procesadas. Este método no requiere conocimiento previo del tipo de señal transmitida, aunque sí de los niveles de ruido presentes en las señales [28].

Al usar detección de energía es muy probable que se tengan señales de falsa alarma si no se conocen bien los niveles de ruido de la señal recibida. Sin embargo, esto podría mejorarse si lo que se usa es una estimación del ruido en la señal y el *Threshold* es dinámico, es decir, se ajusta a los diferentes niveles de ruido. Éste método es tratado en [29].

Detección de energía es el método más usado en radios cognitivos ya que puede ser usado tanto en el dominio del tiempo como de la frecuencia y no requiere de grandes recursos computacionales [30], [31].

Por su parte, el filtro adaptado es un método de procesamiento de señales que consiste en tomar la señal recibida que fue transmitida por un canal AWGN y convolucionarla con otra señal para lograr que su SNR sea reducido, por lo cual es más fácil de identificar. Si se asume presencia de ruido blanco no correlacionado en la señal, la condición para que la reducción del SNR sea máxima, es que la respuesta al impulso de la señal con que se convoluciona la señal recibida sea su conjugada compleja en tiempo invertido. Sin embargo, como es claro, esto requiere de conocimiento previo de la señal recibida y si esta información no es precisa, el método funciona débilmente [32], [33].

El método de detección de característica cicloestacionaria propuesto en [34], parte del hecho de que la mayoría de las señales transmitidas en comunicaciones inalámbricas tienen una base de señales senoidales, trenes de pulsos o contienen prefijos cíclicos, lo que resulta en que exista periodicidad en las señales y esto puede ser usado para diferenciar entre la señal modulada y el ruido. Las señales cicloestacionarias tienen una correlación entre componente muy separadas espectralmente debido a la redundancia espectral causada por la periodicidad. Análogamente a la definición de autocorrelación, se puede definir la función de correlación espectral (SCF) como en la Ecuación (37).

$$S_x^a(f) = \lim_{T \rightarrow \infty} \lim_{\Delta T \rightarrow \infty} \frac{1}{\Delta T} \int_{-\frac{\Delta T}{2}}^{\frac{\Delta T}{2}} \frac{1}{T} X_T \left(t, f + \frac{\alpha}{2} \right) X_T^* \left(t, f - \frac{\alpha}{2} \right) dt \quad (37)$$

Donde α es el parámetro de frecuencia cíclica, T es el periodo de la señal y la transformada de Fourier en tiempo finito está dada por la Ecuación (38).

$$X_T(t, \nu) = \int_{t-\frac{T}{2}}^{t+\frac{T}{2}} x(u) e^{-j2\pi \nu u} du \quad (38)$$

Allí $x(u)$ es la señal y $\nu = f \pm \alpha/2$. La densidad de potencia espectral (PSD) es una transformada unidimensional real-valuada, la SCF es bidimensional y generalmente compleja. La PSD es un caso especial donde $\alpha = 0$. Este método también es trabajado en [35], [36].

El análisis en el dominio del espectro cíclico de la señal, preserva la información de su fase y frecuencia. Diferentes tipos de señales moduladas como BPSK, QPSK y SQPSK que tienen un PSD idéntico tienen, sin embargo, SCF muy diferentes. Además el ruido estacionario no presenta correlación espectral, por lo que la identificación de las señales se facilita y es posible entonces trabajar con señales con bajos niveles de SNR.

La principal desventaja de este método es la complejidad en los cálculos y los recursos de cómputo necesarios para llevar a cabo esta detección en tiempo real [28]. Su ventaja sobre la detección de energía es que permite diferenciar entre diferentes señales o formas de ondas.

Por otro lado, para la técnica de detección basada en autovalores no es necesario conocimiento previo de la forma de la señal o de sus niveles de ruido. Se trata de un método similar al de detección de energía, en donde únicamente se determina si hay presencia de señales enviadas por el usuario primario o no. Se basa en la teoría de matriz aleatoria y consiste en computar la autocorrelación de la señal recibida, definir una matriz de covarianza luego se computa el autovalor máximo y se determina un nivel de referencia *Threshold* con el cual compara para definir si hay presencia o no de señales transmitidas por el usuario primario [37], [32]. Este método también se puede combinar con el tradicional método de detección de energía, como muestra Mohamed Hamid en [38].

En [27] se desarrolla un algoritmo para estimar los parámetros de una señal a partir de métodos de estimación haciendo uso de la función de verosimilitud. En él utilizan funciones *sinc* para modelar la señal recibida y, a lo largo de diferentes iteraciones, se ajustan los parámetros de las funciones *sinc* para lograr una mayor aproximación a la señal original. Los resultados de este algoritmo llamado detección de señal de banda estrecha (*Narrowband Signal Detection*, NSD), son comparados con los resultados del tradicional método de detección de energía y se muestran en la Tabla 1.

Tabla 1. Resultados de algoritmo NSD comparados con Detección de Energía.

SNR	Detección de energía		NSD (<i>Narrowband Signal Detection</i>)	
	P(Detección) [%]	P(Falsa alarma) [%]	P(Detección) [%]	P(Falsa alarma) [%]
0 dB	0.7	98.3	67.3	89.1
5 dB	12.9	97.7	90.1	57.2
10 dB	87.5	70.7	93.9	42.1
15 dB	100	0	94.3	36.7

En la Tabla 1, la primera columna corresponde a los valores SNR con los que se experimentó. Los valores de las columnas 4 y 5 fueron los obtenidos con el algoritmo NSD, mientras que para obtener los valores de las columnas 2 y 3 con el algoritmo de detección de energía, lo que se hizo fue, para la probabilidad de detección, se ajustó un valor de referencia tal que la probabilidad de falsa alarma fuera igual a la obtenida por el algoritmo de NSD para cada caso de SNR. De forma similar se procedió para obtener los datos de probabilidad de falsa alarma. Se puede ver entonces que el algoritmo NSD tiene un mejor desempeño para bajos SNR.

3.2 CLASIFICACIÓN DE MODULACIÓN

En los trabajos consultados en los que se desarrolló un sistema para detectar el tipo de modulación de una señal recibida, se vio que el común entre todos ellos era que hacían uso de algoritmos de *Machine Learning* [27], es decir, algoritmos que aprenden a partir de una lista de datos dada (etapa de entrenamiento) y que luego son capaces de determinar características similares en otras listas de datos o hasta de tomar decisiones respecto a lo aprendido.

En [39] se utilizó un árbol de decisiones para predecir diferentes modulaciones digitales y analógicas, además de clasificar las señales con base en la información de amplitud y fase de las mismas. En este trabajo se encontró que para las características dadas, la razón entre ruido y señal (SNR) debía ser mayor a 10 dB ya que para valores menores, los resultados se veían afectados por el ruido gaussiano con que se realizaron las pruebas. En [40] se compararon diferentes algoritmos: distancia mínima y máquinas de soporte vectorial. Se encontró para el primero un porcentaje de aciertos de 96.2%, mientras que para el segundo algoritmo fue de 98.3%. Ambos algoritmos se probaron para señales con SNR de 5 dB. Algoritmos de redes neuronales fueron puestos a prueba en [41], en donde se usaron modulaciones analógicas y digitales simuladas por computador y el ruido se simuló como un ruido gaussiano de banda limitada. Se encontró que el SNR para el cual el porcentaje de aciertos era superior a 98% es de alrededor de 10 dB. En este trabajo solo se usaron modulaciones con 1 o 2 bits por símbolo.

En [42] se logró identificar el protocolo de comunicaciones Link-4A con aciertos de 91.8% para señales con modulación 2-FSK (*Frequency Shift Keying*) y SNR de -10 dB. Sin

embargo, para el mismo tipo de señal pero con SNR de 5 dB se logró un número de aciertos del 100%. Un árbol de decisiones fue usado en [43] para identificar tres tipos de modulaciones: PAM (*Pulse-Amplitude Modulation*), PSK (*Phase Shift Keying*) y QAM (*Quadrature Amplitude Modulation*). Se identificaron señales con SNR de 6 dB, 8 dB y 10 dB respectivamente para obtener con cada uno de ellos un porcentaje de aciertos de 85%, 90% y 95%. En [44] se utilizaron máquinas de soporte vectorial para identificar modulaciones 2-ASK (*Amplitude Shift Keying*), 4-ASK, QPSK (*Quadrature Phase Shift Keying*), 2-FSK y 4-FSK. Se trabajaron las señales con un SNR de 10 dB y para todas ellas se obtuvo un promedio de 97.5% de aciertos. En [45] se utilizan también máquinas de soporte vectorial, sin embargo aquí se enfocan en realizar una optimización al algoritmo, al probar diferentes *kernels* y determinar cuál de ellos funciona mejor. Se encontró el mejor desempeño para la función de la base radial gaussiana (GRBF) como *kernel* del algoritmo y los parámetros del mismo fueron hallados por medio de un optimizador de enjambre implementado. Una vez escogido y depurado el algoritmo, se probó para diferentes razones señal-ruido: -4 dB, 0 dB, 4 dB, 8 dB, y 12 dB, para los cuales se obtuvieron aciertos de 91%, 94%, 97%, 98% y 99% respectivamente.

Mobasserri plantea en [46] la predicción o reconstrucción del diagrama de constelación de las señales recibidas, para luego determinar el tipo de modulación de las mismas. Para esto, hace uso de agrupaciones Fuzzy que resultan ser suficientes para determinar el diagrama de constelación, a pesar del ruido presente en la señal. Se realizaron tres pruebas, cada una de las cuales usó dos o tres modulaciones simultáneamente y los aciertos fueron de 90% para bajos SNR. Se encontró además que este procedimiento funcionaba bien para sets de datos de cualquier longitud. En [47] utilizaron la transformada Wavelet para filtrar las señales y eliminar el ruido, de esta manera se lograba que el sistema funcionara para SNR bajos. Posteriormente se clasificaban los picos en el diagrama de constelación por medio de un algoritmo de clusterización. Con esto obtuvieron para señales con modulación QAM de 2, 4, 5 y 6 bits por símbolo, aciertos de 72%, 100% y 100% para SNR de 0 dB, 5 dB y 10 dB respectivamente. En [48] se usa el algoritmo de *Machine Learning* conocido como “el vecino más cercano” y en él se usan los datos vecinos para determinar a qué grupo corresponde el dato evaluado. Encontraron que este algoritmo funcionaba bien para SNR bajos, como 3 dB, para lo cual obtuvieron un porcentaje de aciertos superior a 99%.

Según Curtis M. Watson, en [27] los algoritmos de *Machine Learning* más usados para la clasificación y determinación de modulaciones son los árboles de decisión, las redes neuronales y las máquinas de soporte vectorial. Sin embargo, agrega que cada uno de ellos presenta problemas como que los árboles de decisión suelen requerir sets de entrenamiento muy largos, lo cual está limitado por el tipo de problema en que se vaya a aplicar el algoritmo; las redes neuronales se sobreentrenan muy fácilmente y esto implica retrasos en la implementación y, por su parte, las máquinas de soporte vectorial requieren de muchos parámetros de configuración.

CAPÍTULO 4

ESTACIÓN DE TIERRA

El proyecto desarrollado en este trabajo es implementado en el marco de la estación de tierra de la Universidad EAFIT para comunicaciones con elementos de observación de la tierra tales como nano y pico satélites, vehículos aéreos no tripulados (UAVs), drones, globos, cometas y cohetes instrumentados. La estación tiene instalada la capacidad necesaria para establecer comunicaciones *full duplex* en tres bandas: VHF, UHF y Banda S.

En la Figura 5 se muestra un esquema de la estación de tierra, en donde se especifican las referencias de cada uno de los elementos utilizados. La dirección de las flechas indica el flujo de las señales. Todas las conexiones en la estación de tierra se realizaron con cable coaxial RG58.

Todos los elementos de la estación de tierra son controlados desde un computador que se encarga de recibir órdenes acerca de los satélites que debe seguir, hacer su rastreo a través de páginas web, y enviar la posición que deben tomar las antenas para el control del rotor Yaesu G5500. El computador realiza el procesamiento digital de las señales lo cual incluye modulación, codificación, demodulación y decodificación de diferentes modulaciones y protocolos. También se encarga de controlar los estados de los suiches en cada caso de emisión y recepción de las diferentes antenas. El radio Ettus N210 y la *daughterboard* UBX-40 conforman un SDR que se encarga de procesar las señales analógicas y hacer su conversión a digital en el caso de recepción de señales o, en el caso de transmisión de señales realiza una tarea de conversión digital-analógica. A continuación se explica el detalle de las conexiones para ambas configuraciones: recepción y transmisión.

4.1 CONFIGURACIÓN EN RECEPCIÓN

Cuando la señal llega a las antenas es dirigida, en forma de señal eléctrica, a los circuladores pasando primero por los *arresters*, los cuales son elementos de seguridad que protegen toda los instrumentos electrónicos de rayos que impacten en las antenas. Los circuladores se

encargan de dirigir la señal desde la entrada del mismo hasta la salida pero, en caso de que exista una señal que vaya en la dirección contraria, es decir, desde la salida hacia la entrada, el circulador la redirige hacia una salida acoplada que está aislada de la entrada por lo que no se devuelven señales y no se dañan los diferentes elementos electrónicos.

En la estación de tierra se tienen dos circuladores que funcionan en diferentes bandas de frecuencias, estos son el ZFBDC20-13HPS+ para UHF y VHF ya que tiene un rango dinámico que va desde 40 MHz hasta 1000 MHz y soporta potencias de 20 W. El ZGBDC20-33HS+ para Banda S pues su rango dinámico es de 300 – 3000 MHz y puede soportar potencias de hasta 50 W. El hecho de que los amplificadores tengan la capacidad de trabajar en potencias altas es importante dado que por allí circularán las señales amplificadas por los amplificadores de potencia en la etapa de transmisión.

En la configuración de recepción las señales son entonces dirigidas hacia la salida acoplada para luego llegar a un suiche USB-SP4T-63 de Minicircuits. Las entradas de este suiche son las señales recibidas por las tres antenas y, dado que el suiche se puede controlar por protocolo USB, la decisión sobre qué antena va a ser escuchada se toma desde el computador y la velocidad de suicheo no es un problema dado que éste es un suiche de estado sólido y su velocidad de conmutación es de 3 μ s.

Una vez se configure el estado del suiche para escuchar determinada antena, la señal es amplificada por el amplificador de bajo ruido ZX60-33LN+ el cual proporciona una ganancia de alrededor de 15 dB y funciona para todas las antenas puesto que su rango dinámico es de 50 – 3000 MHz. Es importante resaltar que la ganancia del amplificador es diferente para cada frecuencia dentro de su rango dinámico.

Luego del amplificador de bajo ruido, se llega a otro suiche USB-SP4T-63 en donde los estados corresponden a: 1. Recepción de señales, 2. Transmisión de señal para banda UHF y 3. Transmisión de señal para Banda S. En recepción, el puerto común hace contacto con el estado 1 y la señal llega al radio en donde, por medio de un oscilador y un mezclador, se baja la frecuencia de la señal para proceder con la conversión analógica-digital y, finalmente, su procesamiento digital en el computador.

4.2 CONFIGURACIÓN EN TRANSMISIÓN

Para transmisión, el radio recibe las señales moduladas, en baja frecuencia, del computador, hace la conversión digital-analógica, con un oscilador genera una señal a mayor frecuencia y la mezcla con la señal de interés para subir la frecuencia a la deseada. Esta señal llega luego al primer suiche USB-SP4T-63 en donde se decide si se quiere emitir por Banda S o UHF¹. En cualquier caso la señal es procesada por su respectivo amplificador de señal de alta

¹ En el esquema de la Figura 5 se muestra que solo es posible emitir en estas dos bandas, sin embargo la banda VHF quedaría habilitada al mover unos cables o añadir un nuevo suiche.

potencia el cual aumenta la ganancia de la señal que para el caso del amplificador ZHL-5W-1+ son 40 dB y para el ZHL-4W-422+ son 25 dB. La señal luego ingresa al circulador de banda UHF o Banda S, dependiendo el caso, pasa por el *arrester* y finalmente es emitida en forma de radiación por la antena.

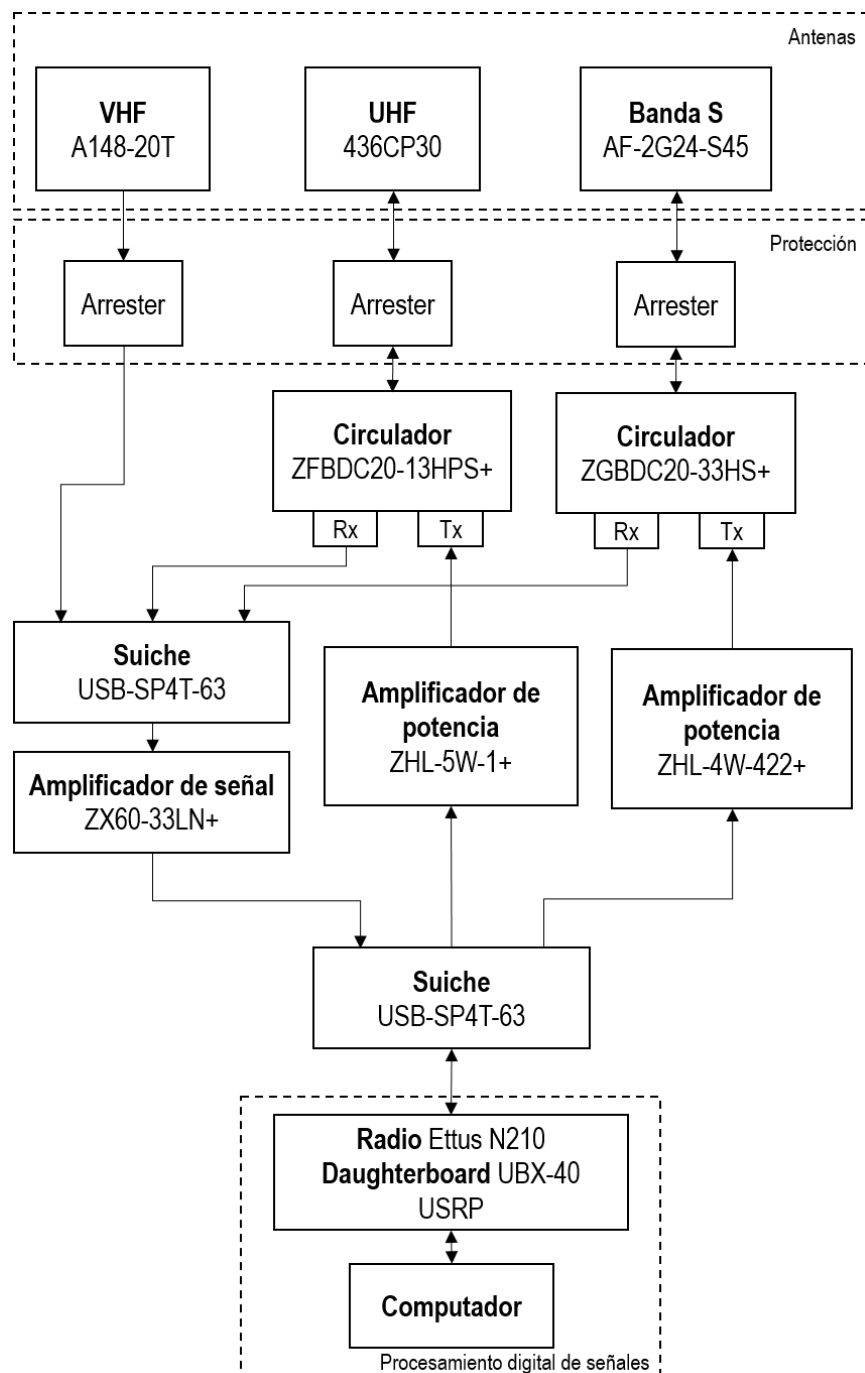


Figura 5. Esquema de la estación de tierra.

4.3 MODELO DE PÉRDIDAS DE LA ESTACIÓN DE TIERRA

En la estación de tierra existen diferentes elementos que aumentan o disminuyen la ganancia de las señales que circulan por ellos. En esta subsección se presentan los cálculos de estas pérdidas y ganancias.

En la Ecuación (39) se presenta el modelo de pérdidas en espacio libre (*Free Space Path Loss*, FSPL) [49].

$$FSPL = \left(\frac{4\pi df}{c} \right)^2. \quad (39)$$

En la Ecuación (39) se tiene que d es la distancia entre el satélite y la estación de tierra, f es la frecuencia de la señal y c es la velocidad de la luz en el vacío. Si se considera que los *CubeSats* se encuentran a unos 700 km de la tierra (el cual es el caso en que la fuente de radiación está más alejada de la estación de tierra) y para una señal VHF con una frecuencia de 140 MHz se tiene

$$FSPL_{VHF} = \left(\frac{4\pi(700 \times 10^3 m)(140 \times 10^6 Hz)}{3 \times 10^8 m/s} \right)^2 = 1.68 \times 10^{13}, \quad (40)$$

$$FSPL_{VHF[dB]} = 10 \log(FSPL_{VHF}) = -132.26 \text{ dB}. \quad (41)$$

Mientras que para una señal UHF con una frecuencia de 440 MHz, se obtiene

$$FSPL_{UHF} = \left(\frac{4\pi(700 \times 10^3 m)(440 \times 10^6 Hz)}{3 \times 10^8 m/s} \right)^2 = 1.66 \times 10^{14}, \quad (42)$$

$$FSPL_{UHF[dB]} = 10 \log(FSPL_{UHF}) = -142.21 \text{ dB}. \quad (43)$$

Para una señal de Banda S con una frecuencia de 2.4 GHz, ocurre lo siguiente

$$FSPL_S = \left(\frac{4\pi(700 \times 10^3 m)(2.4 \times 10^9 Hz)}{3 \times 10^8 m/s} \right)^2 = 4.95 \times 10^{15}, \quad (44)$$

$$FSPL_S[dB] = 10 \log(FSPL_S) = -156.94 \text{ dB}. \quad (45)$$

Se puede ver que las pérdidas para la señal de Banda S son muy grandes, sin embargo esta se expone más como un ejemplo dado que no muchos satélites trabajan en esta banda, debido

precisamente a este tipo de problemas. Con esta banda se espera trabajar con drones, UAVs y otro tipo de elementos de observación de la tierra que no se encuentran a distancias tan grandes.

Ahora bien, las ganancias de las antenas VHF, UHF y Banda S son 13.24 dB, 15.50 dB y 24 dB respectivamente. La ganancia del amplificador de bajo ruido ZX60-33LN+ para la señal VHF es de 21.94 dB, para UHF es de alrededor de 21.51 dB, para Banda S es de 13.11 dB. El cable coaxial usado es RG58 y éste representa diferentes pérdidas para cada frecuencia debido a su longitud, estas son: -20.3 dB/100 m en la banda VHF, -34.8 dB/100m para UHF y -105.6 dB/100 m para Banda S. La totalidad de cable coaxial usado es de unos 30 m para cada antena. El suiche USB-SP4T-63 proporciona pérdidas de alrededor de 1 dB en todas las bandas manejadas y se debe recordar que las señales recibidas deben pasar por dos de estos suiches. Estos datos se resumen en la Tabla 2.

Tabla 2. Ganancias de los diferentes elementos de la estación de tierra.

Ganancias	FSPL [dB]	Antena [dB]	Amplificador [dB]	Cable RG58 [dB/100m]	Suiche [dB]
VHF	-132.26	13.24	21.94	-20.30	-2.00
UHF	-142.21	15.50	21.51	-34.80	-2.00
Banda S	-156.94	24.00	13.11	-105.60	-2.00

Finalmente se obtienen los cálculos totales de pérdidas de las ecuaciones (46), (47) y (48).

$$\begin{aligned} Loss_{VHF} &= -132.26 \text{ dB} + 13.24 \text{ dB} + 21.94 \text{ dB} - 20.3(0.3) \text{ dB} - 2 \text{ dB} \\ &= -105.17 \text{ dB}. \end{aligned} \quad (46)$$

$$\begin{aligned} Loss_{UHF} &= -142.21 \text{ dB} + 15.5 \text{ dB} + 21.51 \text{ dB} - 34.8(0.3) \text{ dB} - 2 \text{ dB} \\ &= -117.64 \text{ dB}. \end{aligned} \quad (47)$$

$$\begin{aligned} Loss_S &= -156.94 \text{ dB} + 24 \text{ dB} + 13.11 \text{ dB} - 105.6(0.3) \text{ dB} - 2 \text{ dB} \\ &= -153.51 \text{ dB}. \end{aligned} \quad (48)$$

4.4 CONTROL DE LA POSICIÓN DE LAS ANTENAS

Como se dijo, se piensa que esta estación de tierra pueda establecer comunicación con nano y pico satélites. Para esto, uno de los puntos importantes es el posicionamiento de las antenas y el seguimiento de las mismas al satélite. Los cálculos requeridos para realizar este seguimiento se desarrollan en esta subsección.

Para posicionar las antenas con el objetivo de hacer seguimiento a un CubeSat mientras éste cruza el cielo dentro del ángulo de visión de las antenas, es necesario conocer la posición en términos de ángulos de latitud y longitud tanto de la estación de tierra como del satélite. Estos datos para la estación de tierra son conocidos, ya que ésta se encuentra en un lugar fijo, mientras que los del *CubeSat* deben ser actualizados constantemente desde una base de datos en internet.

Una vez se conozca la posición de las antenas y el satélite, se tiene la situación mostrada en la Figura 6 en donde ϕ_s es el ángulo de longitud del CubeSat, ϕ_t es la longitud de la estación de tierra, θ_s y θ_t son las latitudes del satélite y la estación de tierra respectivamente, β es el ángulo entre la estación de tierra y el satélite respecto al centro de la Tierra, R es la distancia entre la estación y el centro de la Tierra, R_s es la distancia entre el satélite y el centro de la Tierra, EL es el ángulo de elevación que deben recorrer las antenas para apuntar hacia el satélite.

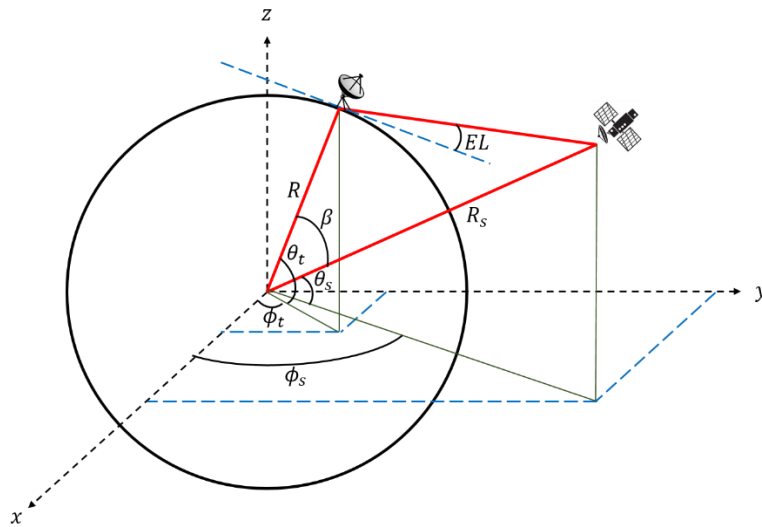


Figura 6. Geometría de la ubicación de un satélite respecto a una estación de tierra.

En la Figura 7 se puede ver el caso de un satélite geoestacionario que, sin pérdida de generalidad, ilustra el cálculo del ángulo azimutal (A) en que se deben mover las antenas de la estación de tierra si estas están inicialmente hacia el norte.

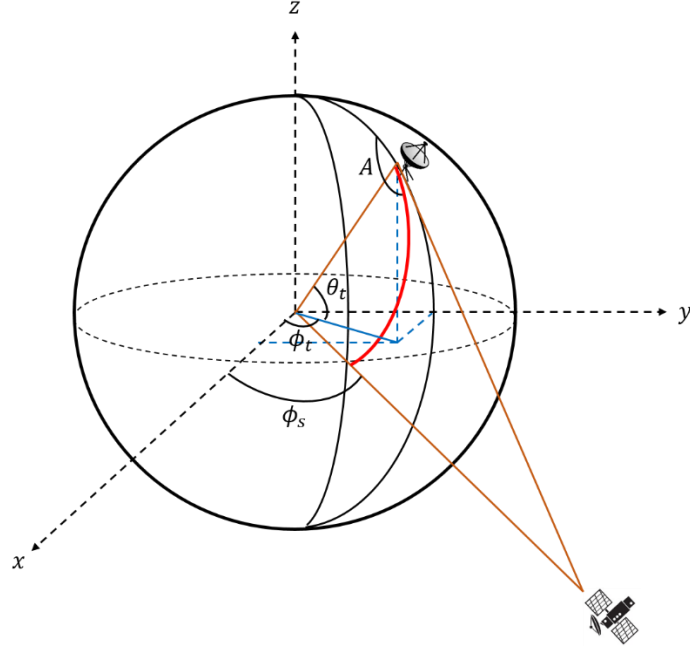


Figura 7. Geometría de la posición de un satélite geoestacionario respecto a una estación de tierra.

Por las leyes de Napier [50], se puede obtener el ángulo azimutal A , como muestra la ecuación (49)

$$A = \tan^{-1} \left(-\frac{\tan(|\phi_t - \phi_s|)}{\sin(\theta_t)} \right). \quad (49)$$

Ahora bien, para el cálculo del ángulo de elevación requerido se tiene para \vec{R} en coordenadas esféricas

$$\vec{R} = R \sin(90^\circ - \theta_t) \cos(\phi_t) \hat{i} + R \sin(90^\circ - \theta_t) \sin(\phi_t) \hat{j} + R \cos(90^\circ - \theta_t) \hat{k} \quad (50)$$

y para \vec{R}_s se tiene

$$\begin{aligned} \vec{R}_s = R_s \sin(90^\circ - \theta_s) \cos(\phi_s) \hat{i} + R_s \sin(90^\circ - \theta_s) \sin(\phi_s) \hat{j} \\ + R_s \cos(90^\circ - \theta_s) \hat{k}. \end{aligned} \quad (51)$$

Normalizando los vectores \vec{R} y \vec{R}_s y tomando su producto punto, se obtiene lo siguiente

$$\begin{aligned} \hat{R} \cdot \hat{R}_s = \sin(\theta_t) \sin(\theta_s) + \cos(\theta_t) \cos(\phi_t) \cos(\theta_s) \cos(\phi_s) \\ + \cos(\theta_t) \sin(\phi_t) \cos(\theta_s) \sin(\phi_s) \end{aligned} \quad (52)$$

y como $\hat{R} \cdot \hat{R}_s = \cos(\beta)$, se obtiene el ángulo β así

$$\beta = \cos^{-1} [\sin(\theta_t) \sin(\theta_s) + \cos(\theta_t) \cos(\theta_s) \cos(\phi_t - \phi_s)] \quad (53)$$

y sea R_{st} la distancia entre el satélite y la estación de tierra, definido así

$$R_{st} = \sqrt{R^2 + R_s^2 - 2RR_s \cos(\beta)}. \quad (54)$$

De esta manera se obtiene finalmente, por ley de senos, el ángulo de elevación de las antenas (EL) que se encuentra en la Ecuación (55)

$$EL = \cos^{-1} \left(\frac{R_s}{R_{st}} \sin(\beta) \right). \quad (55)$$

4.5 MODELACIÓN DE LA ESTACIÓN DE TIERRA

Se realizó una modelación en *SolidWorks* [51] de las antenas de VHF y UHF de la estación de tierra. Esto con el fin de realizar posteriores simulaciones de compatibilidad electromagnética en software de elementos finitos.

La antena VHF usada es la A148-20T de *Cushcraft*. Esta es una antena tipo Yagi de doble polaridad, ganancia de 11.1 dBd y 20 elementos; su rango de trabajo va desde los 144 MHz hasta los 148 MHz y tiene una directividad de 45° en el plano E y 50° en el plano H. Las imágenes de la modelación realizada se pueden ver en la Figura 8, la Figura 9, Figura 10 y la Figura 11.

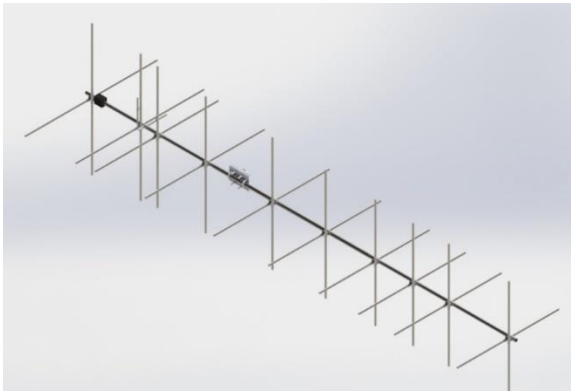


Figura 8. Vista isométrica de la antena VHF.

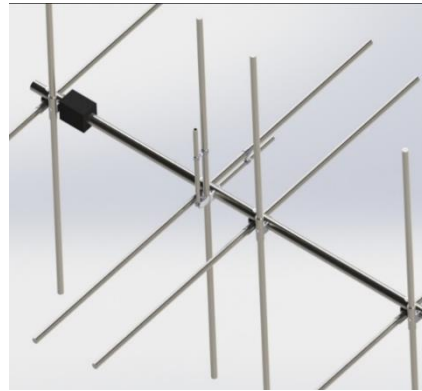


Figura 9. Detalle de la antena VHF.

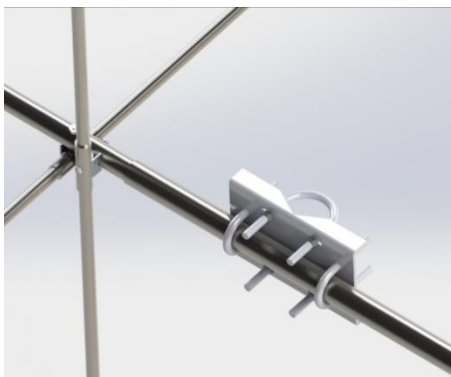


Figura 10. Detalle de la sujeción al mástil de la antena VHF.



Figura 11. Detalle de las sujeciones de los elementos en la antena VHF.

La antena UHF de la estación de tierra es una 436CP30 de *M2 Antenna Systems*. Esta también es una antena tipo Yagi con doble polaridad, 15.5 dBi de ganancia y 30 elementos. Tiene una directividad circular con apertura de 30°. El rango de frecuencias en que trabaja es desde 432 MHz hasta 440 MHz. La Figura 12, la Figura 13 y la Figura 14 muestran los detalles de la modelación realizada.

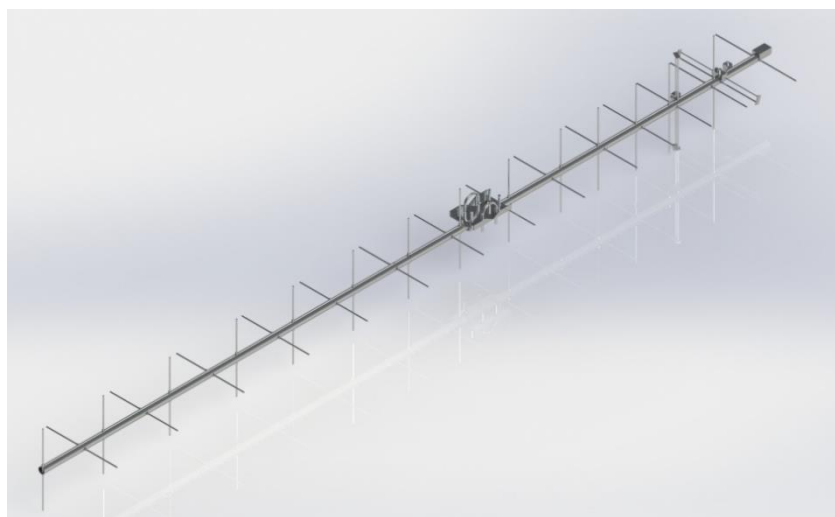


Figura 12. Vista isométrica de la antena UHF.

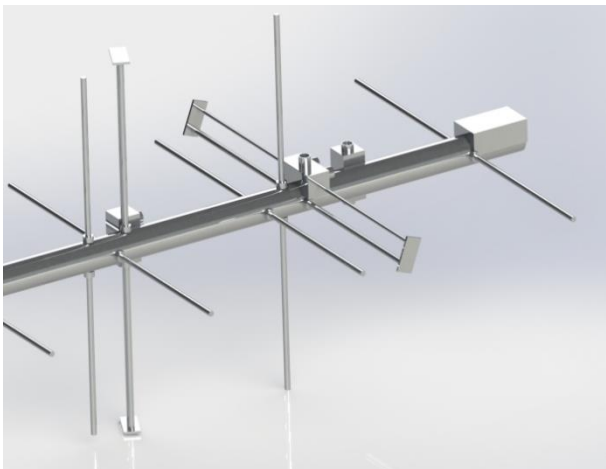


Figura 13. Detalle de la antena UHF.

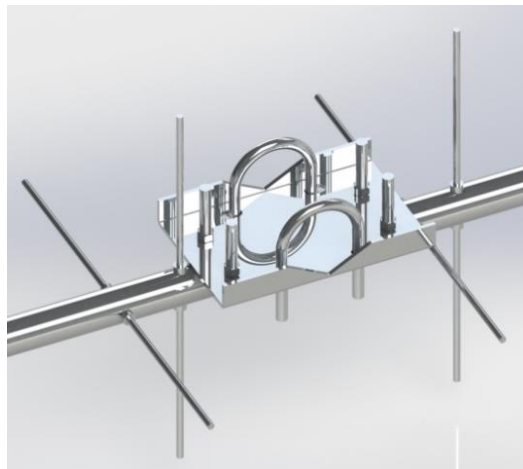


Figura 14. Detalle de la sujeción al mástil de la antena UHF.

CAPÍTULO 5

METODOLOGÍA

5.1 CARACTERIZACIÓN DE LA POTENCIA DE RECEPCIÓN DEL RADIO

Para determinar la amplitud de la señal recibida por el radio, es necesario primero realizar una caracterización del mismo ya que, luego de un análisis en transformada de Fourier, la amplitud es dada en decibeles, lo que no es una medida absoluta de la potencia de la señal entrante. Esta medida además, depende tanto de la frecuencia de la señal, como de la ganancia de recepción, como se verá en este capítulo.

5.1.1 Montaje

Se caracterizó el radio Ettus N210 [52] con un modulador UBX40 [53] en cual tiene 40 MHz de ancho de banda y trabaja en frecuencias desde 10 MHz hasta 6 GHz; para esto se utilizaron dos antenas UHF de dipolo convencionales para transmisión y recepción; un medidor de potencia del fabricante *Mini-Circuits* de referencia ZX47-60LN-S+ [54] y un Arduino UNO para la adquisición de datos y el envío de los mismos al computador por comunicación serial. La Figura 15 muestra el montaje realizado.

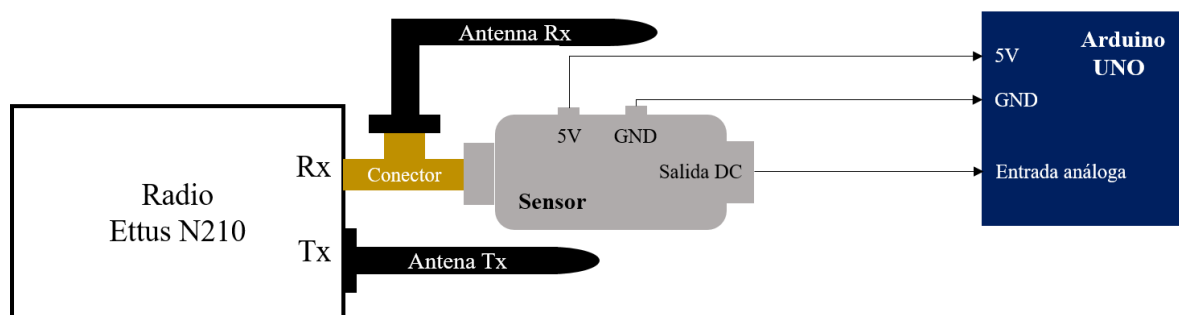


Figura 15. Montaje para la caracterización de la potencia de recepción.

En la conexión del puerto de Rx del radio se usó un conector en T para que la potencia recibida por la antena Rx fuera distribuida uniformemente entre el sensor de potencia y la entrada del radio.

5.1.2 Toma de datos

Se escribió un algoritmo en Python 2.7 usando GNU Radio para controlar el radio y hacer que se barriera un cierto ancho de banda con un paso en frecuencias determinado y que, en cada paso, se cambiara la ganancia de transmisión desde 0 dB hasta 20 dB. En cada uno de esos pasos se hace una toma de datos de la señal recibida, de la cual se toma su transformada de Fourier y el punto máximo de la misma para determinar la amplitud de la señal recibida, en términos del algoritmo de estimación de parámetros propuesto en este trabajo. Una vez se ha realizado el barrido de frecuencias y ganancias, se guardan los datos calculados en un archivo binario tipo numpy al cual tiene una extensión npy.

Por otra parte, hay un Arduino Uno que recibe la señal análoga correspondiente a las medidas del sensor de potencia. Se escribió un algoritmo para el Arduino en donde se toman diez datos recibidos por el puerto análogo y se saca un promedio de los mismos para luego mandar este promedio por comunicación serial a un computador. Se decidió sacar un promedio de estos datos ya que la señal del sensor no es muy estable.

También se escribió otro algoritmo en Python 2.7 para interpretar los datos recibidos por el puerto serial del computador. En este algoritmo, se toman los datos, se convierten a valores de voltaje y luego, por medio de la caracterización del sensor de potencia, dada por el fabricante, se convierten a potencia en dBm. Estos datos finalmente son guardados en un archivo tipo numpy.

Finalmente, los archivos que contienen los datos de potencia tanto del radio como del sensor, son confrontados y, puesto que se vio que los datos del sensor son ruidosos, se decidió hacer una regresión lineal de los datos para aproximar y de esta manera facilitar las interpolaciones y extrapolaciones que sean necesarias en etapas posteriores.

La caracterización fue realizada solo para las bandas de frecuencias disponibles actualmente en la estación de tierra de la Universidad EAFIT, las cuales son VHF, UHF y Banda S. Para la banda VHF se caracterizó desde 144 MHz hasta 148 MHz con un paso de 50 kHz. Para UHF se empezó desde 430 MHz y se terminó en 440 MHz con un paso de 50 kHz, mientras que para Banda S se hizo un barrido desde 2.4 GHz hasta 2.5 GHz con pasos de 1 MHz.

5.1.3 Resultados

Los datos correspondientes a algunas de las frecuencias evaluadas y sus correspondientes aproximaciones por regresión lineal, son mostrados en la Figura 16, Figura 17 y Figura 18.

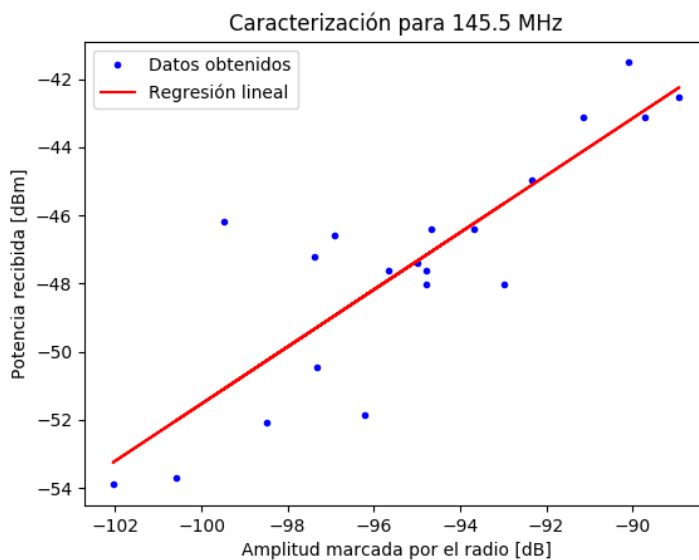


Figura 16. Datos de la caracterización y la aproximación por regresión lineal en 145.5 MHz

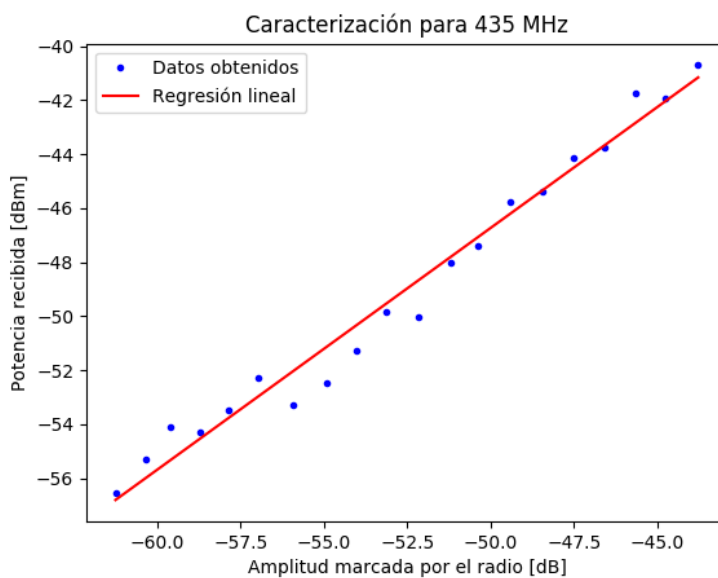


Figura 17. Datos de la caracterización y la aproximación por regresión lineal en 435 MHz

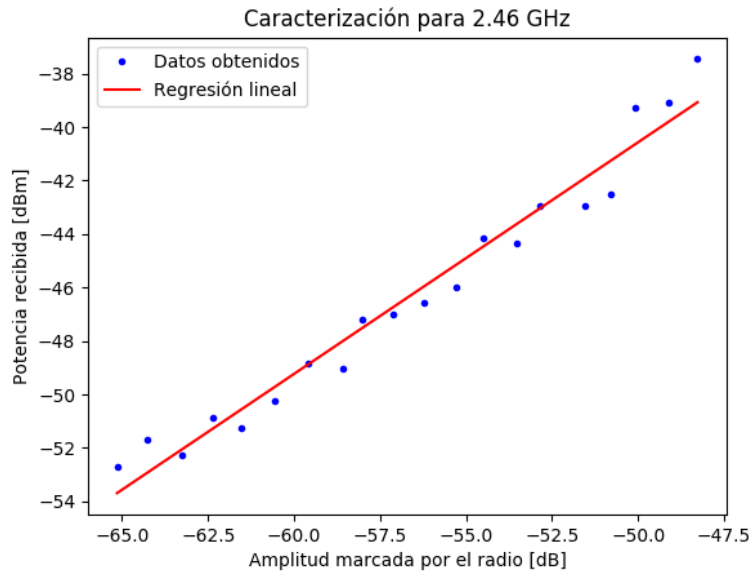


Figura 18. Datos de la caracterización y la aproximación por regresión lineal en 2.46 GHz.

Con esta caracterización se verificó que efectivamente los datos de potencia de recepción del radio dependen de la frecuencia de la señal recibida y esto se puede ver con claridad en la Figura 19, en donde se graficó la relación entre la amplitud del radio y la potencia real a diferentes frecuencias. Allí se puede ver que ninguna curva es un múltiplo de la otra, por lo que se confirma la importancia de la caracterización hecha.

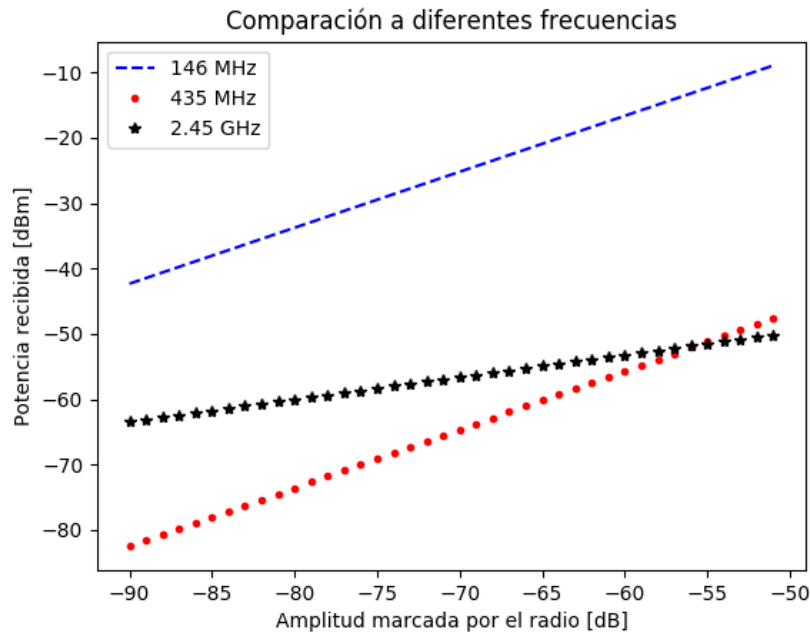


Figura 19. Relación entre la amplitud marcada por el radio y la potencia de la señal, a diferentes frecuencias.

El archivo que contiene los datos correspondientes a esta caracterización tiene una extensión .npy y tiene un peso de 199 KB por lo que no representa un gran problema para el tiempo de cómputo del algoritmo de estimación de parámetros.

5.2 CREACIÓN DE LOS SETS DE DATOS

Una parte de los datos utilizados para el entrenamiento de los algoritmos fueron datos simulados, es decir, fueron creados como conjuntos de datos en el computador y los ambientes de transmisión y canal fueron simulados matemáticamente.

5.2.1 BPSK

Para generar las señales con modulación BPSK se utilizaron los algoritmos desarrollados por Tim O'Shea en [55]. En ellos existe la posibilidad de generar sets de datos con diferentes modulaciones, sin embargo solo se utilizó para BPSK. Allí se usa un fragmento de los sonetos de Shakespeare en ASCII como fuente de alfabeto y, de esta forma, asegurar que los bits aparecen de forma equiprobable. Estos bits son posteriormente interpolados de manera que se alcance el número de muestras por símbolo deseado y se les da una modulación, que este caso corresponde a BPSK. Como último paso para simular la etapa de transmisión, se le aplica a la señal un filtro de forma de raíz de coseno para limitar el ancho de banda de la señal. Luego la señal procesada por el bloque de canal dinámico de GNU Radio el cual incluye efectos como cambios en la frecuencia de muestreo de la señal, cambios de la frecuencia central, un modelo selectivo de *fading* y un modelo de ruido blanco gaussiano. Para simular el cambio en la frecuencia de muestreo se usa un interpolador en donde, a partir de una desviación determinada (*Sample rate offset*, SRO), se interpola la señal portadora. El modelo con el cual se cambia la frecuencia central de la señal en el canal (*center frequency offset*, CFO) consta de un oscilador y un mezclador que crea una señal a una frecuencia determinada y la mezcla con la señal en canal para obtener un cambio de frecuencia en la portadora. Para el modelo selectivo de *fading* se usa el método de suma de señales sinusoidales con un ruido de fase aleatorio para simular los procesos de *fading* de *Rician* o *Rayleigh*. Para el ruido blanco se usa el proceso normal AWGN en donde la señal es distorsionada por un ruido blanco de acuerdo al SNR deseado. Éste último proceso simula las condiciones térmicas del canal.

Los valores usados para el bloque de canal dinámico se listan en la Tabla 3. Cabe resaltar que para la generación de datos se usaron diferentes valores de SNR. Estos se variaron desde -20 dB hasta 18 dB en pasos de 2 dB.

Tabla 3. Parámetros del bloque de canal dinámico de GNU Radio.

Parámetro	Valor
Frecuencia de muestreo	200 000 Hz
Desviación estándar de la frecuencia de muestreo por muestra	0.01 Hz
Desviación máxima de la frecuencia de muestreo	100 Hz
Desviación estándar de la frecuencia central por muestra	0.01 Hz
Desviación máxima de la frecuencia central	1 000 Hz
Número de señales sinusoidales usadas en el modelo selectivo de <i>fading</i>	8
Máxima frecuencia Doppler usada en el modelo selectivo de <i>fading</i>	1 Hz
Factor K que define la razón de potencia entre canal directo y difuso	4
Listado de retardos que conforman el perfil de retardo	[0.0, 0.9, 1.7]
Listado de magnitudes correspondiente a los retardos anteriores	[1, 0.8, 0.3]
Número de taps del filtro para interpolar el perfil de retardo de potencia	8
Desviación estándar para el proceso AWGN	$10^{-(\text{SNR}/10)}$
Número que sirve como semilla para la fuente de ruido aleatorio	0x1337

De esta forma de crearon 500 sets de datos en modulación BPSK, en donde cada set consta de 1024 datos que reconstruyen la señal en el tiempo. Esta base de datos se guardó en un archivo .npz en forma de matriz 2D con dimensiones (500 x 1025). Allí la primera columna es el valor SNR con el cual fue calculada la señal que se encuentra la fila correspondiente.

En la Figura 20, Figura 21 y Figura 22, se pueden ver las señales BPSK generadas para valores SNR de -20 dB, 0 dB y 18 dB respectivamente. Allí el eje horizontal representa simplemente la posición de cada dato en el vector y se puede ver que se tomó una ventana de 500 datos. El eje vertical es una amplitud en unidades arbitrarias.

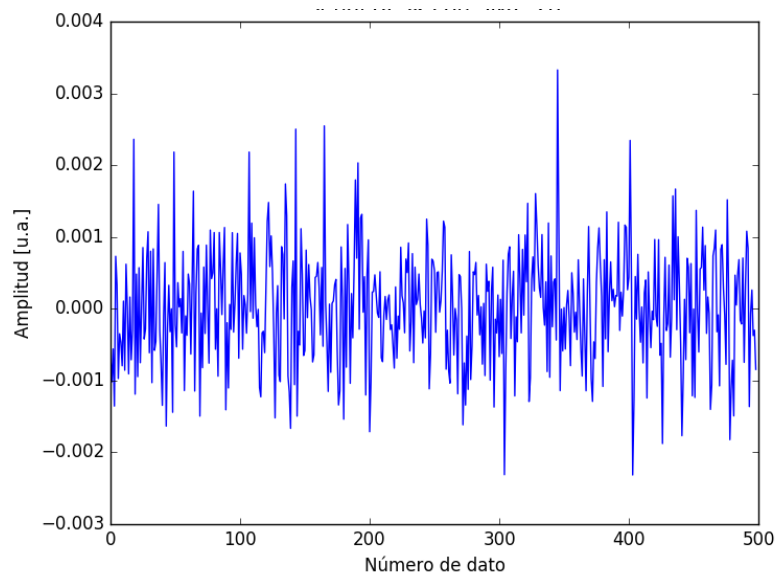


Figura 20. Señal BPSK generada con SNR de -20 dB.

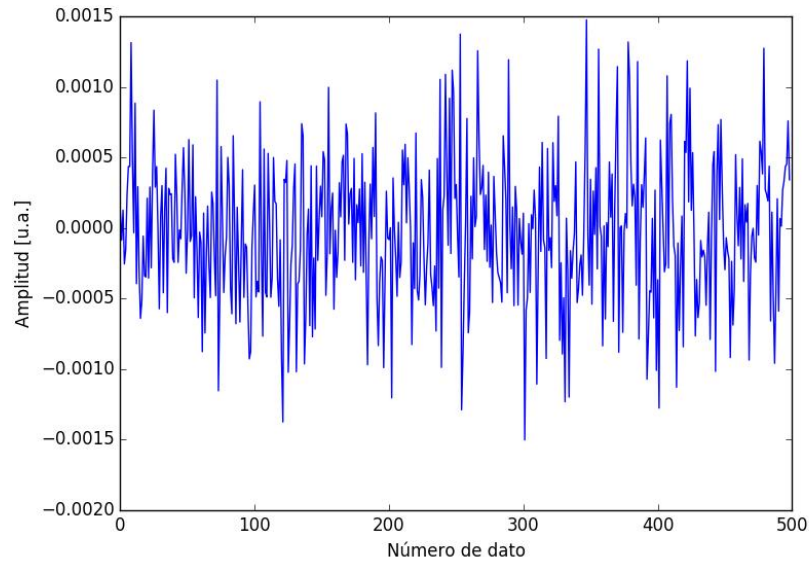


Figura 21. Señal BPSK generada con SNR de 0 dB.

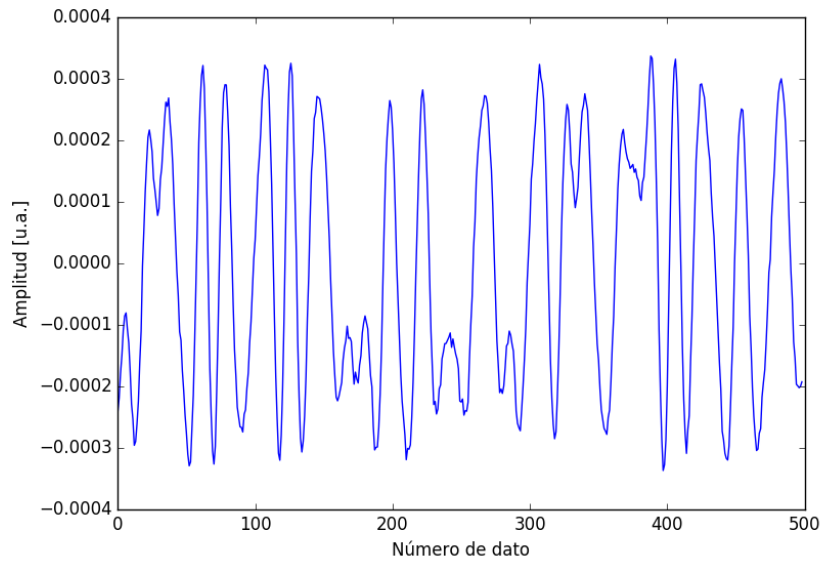


Figura 22. Señal BPSK generada con SNR de 18 dB.

5.2.2 BFSK

Para generar la base de datos de señales con modulación BFSK se utilizó como base el mismo algoritmo utilizado para las señales BPSK y solo se modificó la forma de modular la señal. Para esto se utilizaron las Ecuaciones (56) y (57) en caso de que el bit presente en la información fuera un 1 o un 0, respectivamente.

$$s_0(t) = A \cos(2\pi t(f_c + f_0)), \quad (56)$$

$$s_1(t) = A \cos(2\pi t(f_c + f_1)). \quad (57)$$

Allí A es la amplitud de la señal que fue fijada como 1, t corresponde al tiempo transcurrido en un periodo de símbolo, en este caso el periodo de símbolo fue de $1/br$ donde br es la tasa de símbolo que fue definida como $br = 1200 \text{ Hz}$. Aquí t es un vector de 8 posiciones dado que 8 fue el número de muestras por símbolo. f_c es la frecuencia de la señal portadora. Finalmente f_0 y f_1 son las frecuencias correspondientes a los bits 0 y 1 respectivamente.

El modelo de canal usado fue el mismo descrito para los sets de datos en BPSK y los parámetros fueron los mismos. Finalmente se obtuvo un archivo `.npz` a partir de una matriz 2D de dimensiones (500 x 1025) en donde existen 500 señales en el tiempo con modulación BFSK a diferentes SNR, de tal manera que hay 25 señales para cada valor de SNR que varía desde -20 dB hasta 18 dB en pasos de 2 dB. Cada señal contiene 1024 datos y la primera columna de los datos contiene el valor de SNR con el cual fue calculada la señal que se encuentra en la fila correspondiente.

En la Figura 23, Figura 24 y Figura 25, se pueden ver respetivamente las señales BFSK generadas para valores SNR de -20 dB, 0 dB y 18 dB. En dichas figuras el eje horizontal representa la posición de cada dato en el vector y se puede ver que se tomó una ventana de 400 datos. El eje vertical es una amplitud en unidades arbitrarias.

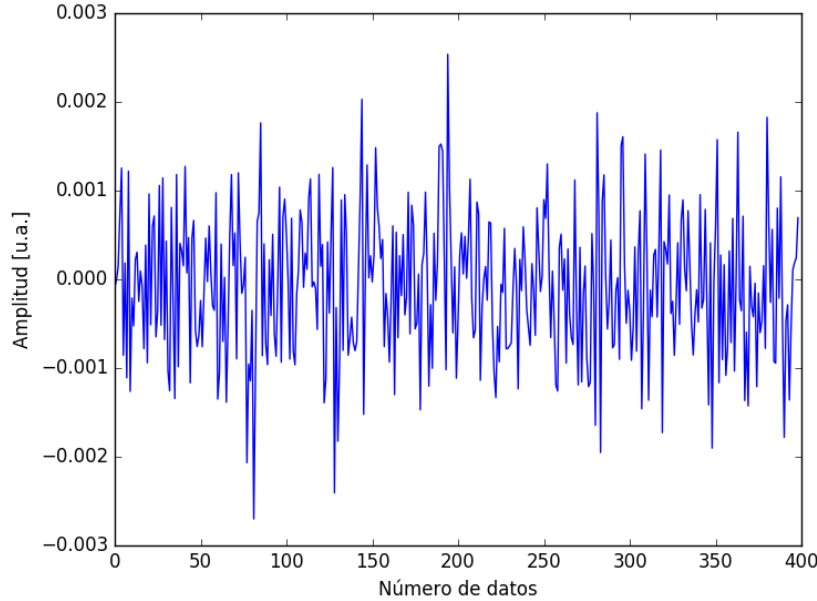


Figura 23. Señal BFSK generada con SNR de -20 dB.

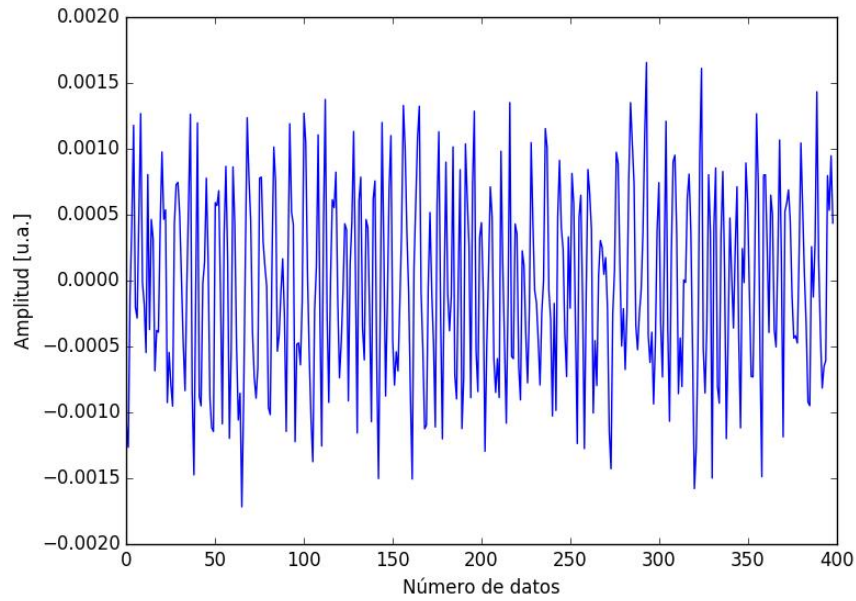


Figura 24. Señal BFSK generada con SNR de 0 dB.

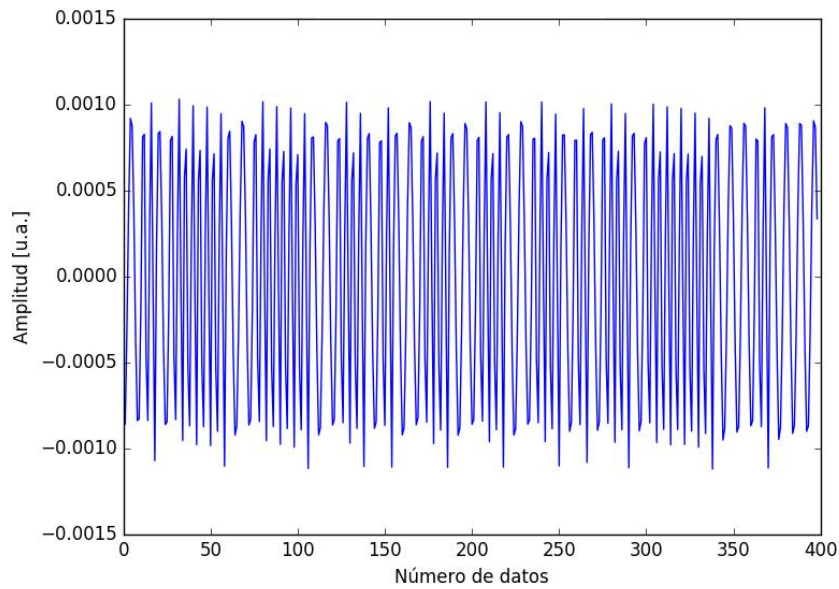


Figura 25. Señal BFSK generada con SNR de 18 dB.

5.2.3 Señales capturadas con el SDR

También se utilizaron algunos sets de datos capturados por el SDR para hacer pruebas de los algoritmos con señales reales. Estas señales fueron generadas por dispositivos comerciales de bajo costo que emiten a diferentes frecuencias como en las bandas VHF, UHF y Banda S.

Con estos, se generaron tres sets de datos que contenían 200 señales en el tiempo y con las cuales se probó posteriormente el algoritmo de estimación de parámetros. Estas señales se pueden ver en la Figura 26, la Figura 27 y la Figura 28.

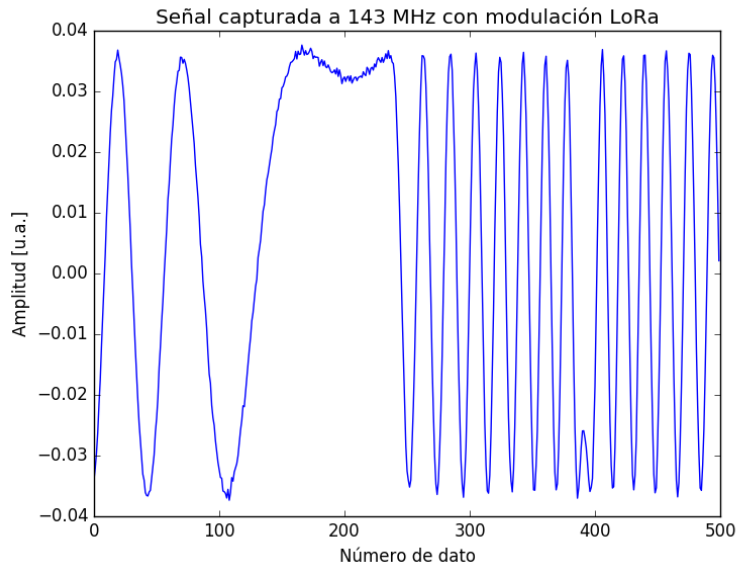


Figura 26. Señal real en VHF con modulación LoRa.

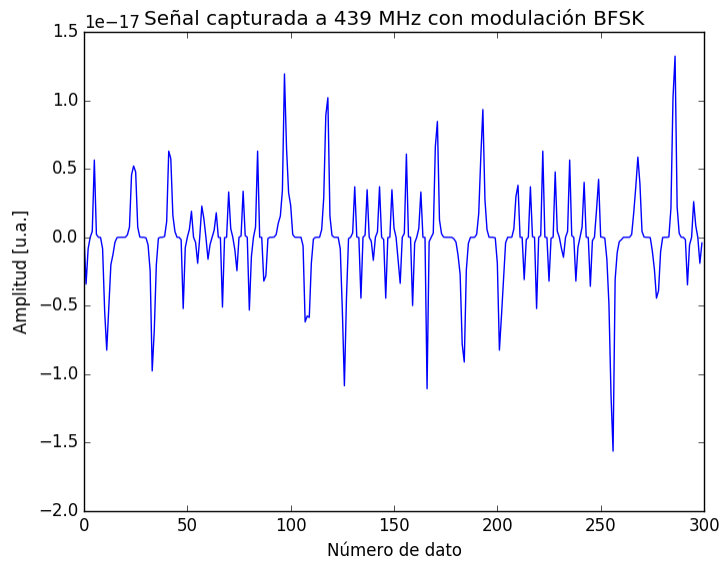


Figura 27. Señal real en UHF tomada con un offset de 25000 Hz y con modulación FSK.

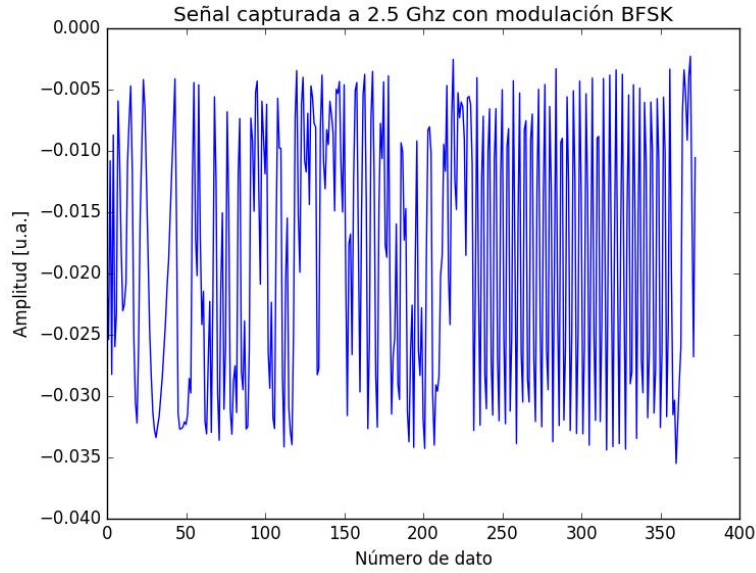


Figura 28. Señal real en Banda S con modulación FSK.

5.3 ALGORITMO DE ESTIMACIÓN DE PARÁMETROS

El algoritmo de estimación de parámetros desarrollado se encarga de tomar una señal muestreada en el tiempo y determinar su frecuencia central, amplitud y ancho de banda a partir del cálculo de la transformada de Fourier discreta de la señal y de un método de aproximación iterativo.

Como se dijo, el primer paso del algoritmo es el cálculo de la transformada de Fourier de señal que, una vez obtenida, es quien va a ser aproximada por una función de valor absoluto de *sinc*.

$$s(x) = \left| A \operatorname{sinc} \left(\alpha (x - x_{\text{offset}}) \right) \right|. \quad (58)$$

En la Ecuación (58), A es la amplitud de la función *sinc*, α es un parámetro para el ancho del lóbulo central de la función y x_{offset} es el punto en el eje x en donde se centra la función. La razón por la que la aproximación se hace con una función *sinc* es porque en las comunicaciones digitales es común usar filtros de forma para las señales que serán transmitidas y uno de los filtros más usados para este fin es el filtro *sinc* por lo que las señales recibidas suelen tener esta forma en su transformada de Fourier. Existen otros tipos de filtros de forma como el de raíz de coseno, sin embargo se podrá ver que para el propósito de este

trabajo que es dar una estimación de los parámetros mencionados, la aproximación con función *sinc* puede ser suficiente.

El algoritmo se divide en tres subrutinas cada una de las cuales se ocupa de un parámetro diferente. En las secciones siguientes se explicará cada una de las subrutinas.

5.3.1 Amplitud

La aproximación inicial de la amplitud realizada por el algoritmo es tomar el valor máximo de la transformada discreta de Fourier de la señal de entrada, por lo tanto si F es esa transformada, se tiene que

$$A = \max(F) . \quad (59)$$

De hecho en ese mismo punto en donde se encontró el máximo de la transformada se fija el offset inicial de la aproximación, así

$$x_{offset} = F^{-1}(\max(F)) . \quad (60)$$

Ahora se determina una función de error sobre la cual se optimizarán los valores de amplitud de la *sinc*. Para esto se define primero una función residuo V que es la resta entre la transformada de Fourier de la señal original y la aproximación realizada. El error se calcula entonces como un error cuadrático, así [27]

$$E = \sum_{l \in L} (V_l - \Delta_A s(x_l))^2 . \quad (61)$$

Donde L es el intervalo de puntos en x contenidos en el lóbulo central de la *sinc* y Δ_A es el parámetro sobre el que se debe optimizar para hallar el cambio de amplitud que debe ser aplicado con el fin de reducir el error. Al derivar la Ecuación (61) respecto a Δ_A e igualar a cero se obtiene

$$\Delta_A = \frac{\sum_{l \in L} [V_l s(x_l)]}{\sum_{l \in L} [s(x_l)^2]} . \quad (62)$$

Una vez obtenido Δ_A se procede a actualizar el valor de la amplitud de la función *sinc*. Es bueno que el valor calculado no se haga efectivo inmediatamente porque eso puede crear cambios muy grandes en la amplitud y por lo tanto divergencias en el algoritmo. Para evitar eso se hace un *smooth* por medio de un parámetro λ al que se le varía de 0 a 1 hasta encontrar un valor que haga que el algoritmo no diverja.

$$A = (1 - \lambda)A + \lambda \Delta_A \quad (63)$$

Para esta subrutina se calculó un error cuadrático medio entre la transformada de Fourier de la señal evaluada y la aproximación con valor absoluto de *sinc*. Este error es calculado sólo sobre los valores contenidos en el lóbulo central de la aproximación *sinc*. El error es calculado en cada iteración y, si este alcanza un valor inferior o igual a 10, la rutina para y entrega la aproximación actualizada.

Sin embargo, existen casos en que, por el ruido de las señales, el error nunca es menor a 10 por lo que se hacen muchas iteraciones y finalmente retorna una amplitud no válida *nan* (*not a number*). Para evitar que esto represente un problema para las subrutinas siguientes, se incluyeron dos casos:

- Si la subrutina ha corrido diez veces, se retorna al valor inicial que es aquel en el que la amplitud de la aproximación es igual al valor máximo de la transformada de Fourier de la señal.
- Si al finalizar la subrutina se tiene un valor de amplitud *nan*, se retorna también al valor inicial.

5.3.2 Frecuencia central

Como se dijo en la sección anterior, la primera aproximación de la frecuencia central se hace en el punto en donde está el valor máximo de la transformada de Fourier de la señal, como muestra la Ecuación (60). El modelo usado para el cálculo de los cambios que se deben dar en la frecuencia central, es un modelo de centro de masa [27] en donde se calcula la diferencia entre la función *sinc* y la función residuo para luego hacer un balance de torques usando como pivote x_{offset} y encontrar el centro de masa. De esta forma se definen L_R y L_L .

$$L_R = \{l : l \in L, l < x_{offset}\} \quad (64)$$

$$L_L = \{l : l \in L, l \geq x_{offset}\} \quad (65)$$

Luego se calculan los aportes de cada uno de los lados, de la siguiente manera

$$M_R = \frac{1}{L} \sum_{l \in L_R} [V_l - s(x_l)] \quad (66)$$

$$M_L = \frac{1}{L} \sum_{l \in L_L} [V_l - s(x_l)] \quad (67)$$

Finalmente se compara M_R con M_L y si el primero es mayor al segundo, el offset debe ser movido hacia la derecha y viceversa. Para esto se usó una función `clip()` que limita el cambio en el offset a solo una unidad en cada iteración dado que cambios muy grandes producen que el algoritmo no converja.

El criterio de parada de esta subrutina es que si, en dos iteraciones consecutivas, el offset se movió una unidad en una dirección y luego una unidad es la otra dirección, quiere decir que ya convergió a una solución por lo que para y retorna la función *sinc* actualizada.

5.3.3 Ancho de banda

Para el planteamiento de la función del error con el cual se puede calcular el cambio del ancho de banda en cada iteración, se debe hacer algo diferente a lo que se hizo para la frecuencia central y es que si se plantea el error como la resta entre V y $s(\alpha, x)$ y luego se deriva respecto a un Δ_α no es posible encontrar una expresión analítica para despejar Δ_α por lo cual se recurre a una nueva aproximación en donde se quiere aproximar el lóbulo central de la función *sinc* con la expresión cuadrática de la Ecuación (68) [27].

$$Q(\alpha, x) = A - A \left(\frac{\Delta_\alpha}{L} \right)^2 (x - x_{offset})^2. \quad (68)$$

Se debe tener en cuenta que la aproximación anterior se obtuvo de manera tal que, al evaluar $Q(\alpha, \varphi)$ se obtenga V , dado que

$$\varphi = x_{offset} + \frac{\mu L}{\Delta_\alpha} \sqrt{\frac{A - V}{A}}. \quad (69)$$

En la expresión anterior se usó el parámetro μ el cual está definido como sigue

$$\mu = \begin{cases} 1 & \text{si } x > x_{offset} \\ -1 & \text{si } x \leq x_{offset} \end{cases}. \quad (70)$$

Ahora a partir de este punto se puede definir una función de error y de hecho se pueden definir dos funciones de error como se muestra [27]

$$E_V = \sum_{l \in L} (V_l - Q(\alpha, x_l))^2. \quad (71)$$

$$E_H = \sum_{l \in L} (x_l - \varphi_l)^2. \quad (72)$$

Los errores E_V y E_H corresponden a errores vertical y horizontal, respectivamente. Ambos toman valores muy diferentes dentro del algoritmo y resulta de utilidad evaluar cuál es más conveniente usar o sacar un valor ponderado de los dos.

Al integrar los dos errores, igualar a cero y despejar Δ_α para cada caso se obtiene

$$\Delta_{\alpha,V} = \frac{\sum_{l \in L} [(A - V_l)(x_l - x_{offset})^2]}{\sum_{l \in L} \left[\frac{A}{L} (x_l - x_{offset})^4 \right]}, \quad (73)$$

$$\Delta_{\alpha,H} = \frac{\sum_{l \in L} [\Gamma_l^2]}{\sum_{l \in L} [(x_l - x_{offset}) \mu \Gamma_l^4]}, \quad (74)$$

donde $\Gamma_l = L \sqrt{\frac{A - V_l}{A}}$.

El criterio de parada de esta subrutina también se definió con base en el error cuadrático medio de la transformada de Fourier de la señal evaluada y la aproximación *sinc*, evaluado en el lóbulo central. En este caso la subrutina hace 50 iteraciones y en cada una de ellas guarda en diferentes vectores los valores de error cuadrático medio y ancho de banda. Luego de correr este número de veces, se toma el valor de ancho de banda para el cual el error fue mínimo. Con esto se asegura que la subrutina converja siempre a un valor.

El algoritmo de estimación de parámetros corre las subrutinas anteriores de forma iterativa hasta alcanzar una aproximación con un error cuadrático medio bajo. Este error es calculado en el lóbulo central de la aproximación.

El valor de amplitud entregado es calculado a partir de la caracterización hecha para la potencia de recepción del radio en el caso en que se trabajan con señales reales adquiridas por el mismo.

Este algoritmo cuenta con un paso de pre-procesamiento en donde se realiza una primera aproximación de la transformada de Fourier de la señal recibida. Aquí se hace una aproximación por picos en donde, en una ventana de datos determinada (que para los valores mostrados se mantuvo en 20 datos), se determinan los dos valores más altos o picos y se realiza una interpolación lineal entre dichos valores. Finalmente se reúnen las interpolaciones realizadas y la función resultante es quien realmente es aproximada por la función *sinc* en las subrutinas descritas anteriormente.

Para las señales BFSK es necesario que el algoritmo genere dos aproximaciones dado que estas señales tienen, en su transformada de Fourier, dos picos que representan las frecuencias principales. El algoritmo está limitado a solo dos aproximaciones, es decir, si se da el caso

en el que tres señales diferentes están presentes en el ancho de banda sensed, el algoritmo solo detectará dos señales. En el caso en que las dos aproximaciones arrojadas tengan sus frecuencias centrales ubicadas a menos de 50 kHz se considera que ambas aproximaciones hacen parte de la misma señal y por lo tanto el ancho de banda que se entrega es el de la suma de los anchos de banda calculados con las aproximaciones.

A continuación, se muestra un esquema del algoritmo de estimación de parámetros en donde son claros los ciclos utilizados en el algoritmo.

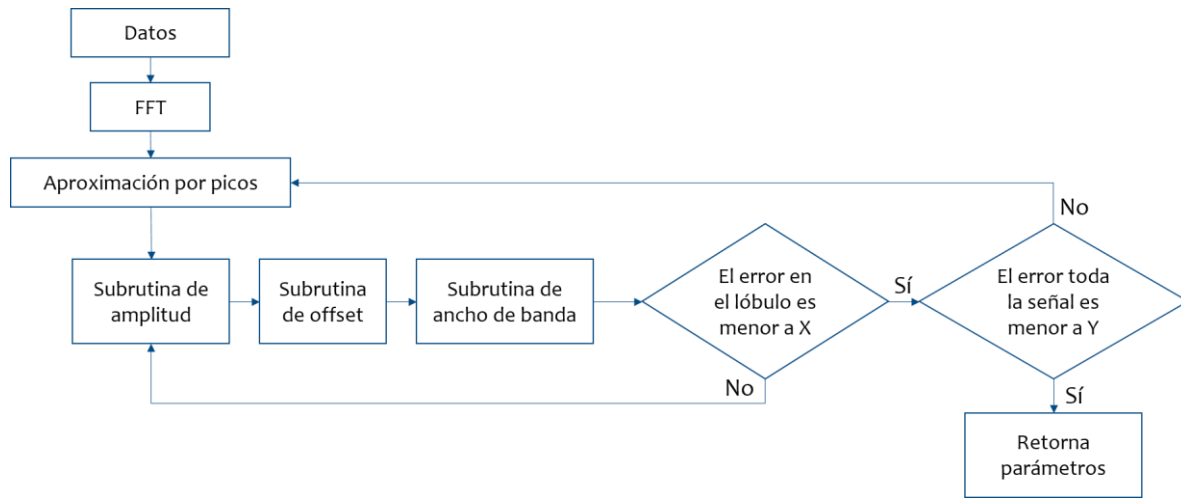


Figura 29. Esquema del algoritmo de estimación de parámetros.

5.4 CLASIFICACIÓN DE MODULACIÓN

Para la clasificación de modulación se usaron algoritmos de redes neuronales, máquinas de soporte vectorial y árboles de decisión que fueron entrenados para aprender diferentes características de las modulaciones BPSK y BFSK para que luego pudiera predecir la modulación de otras señales diferentes a las usadas durante el entrenamiento.

Se usaron diez tipos diferentes para representar las modulaciones usadas, todas ellas basadas en el cálculo de la desviación estándar de la fase y de la frecuencia instantánea de la señal evaluada. Este cálculo para cada uno de estos parámetros se realiza como muestra la Ecuación (75) [56].

$$\sigma = \sqrt{\frac{1}{L} \left[\sum \varphi^2(i) \right] + \left[\frac{1}{L} \sum \varphi^2(i) \right]^2}, \quad (75)$$

Donde L es el número de muestras y φ es la fase o la frecuencia instantánea de la señal. La frecuencia instantánea se define como muestra la ecuación (76).

$$\varphi_f(t) = \frac{1}{2\pi} \frac{d[\arg(z(t))]}{dt} \quad (76)$$

Para la fase, la función $\varphi(t)$ se define de la siguiente forma

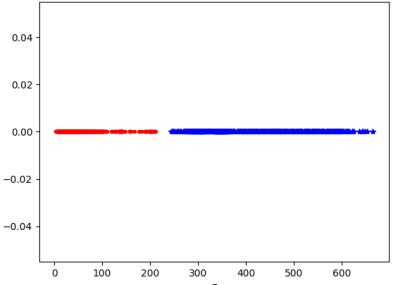
$$\varphi_p(t) = \text{unwrap}(\arg(z(t))). \quad (77)$$

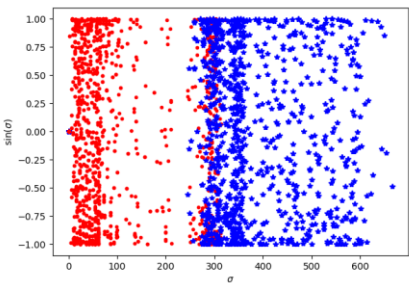
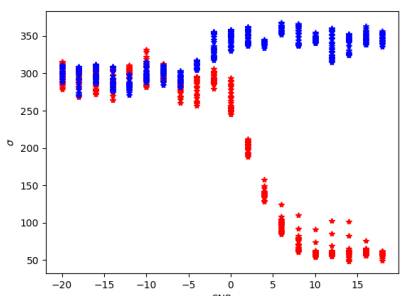
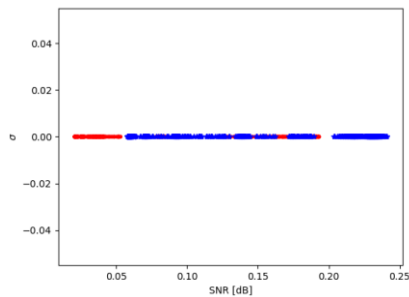
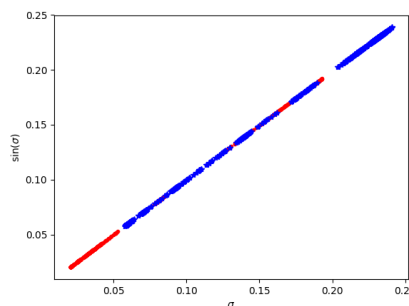
Allí se utilizó $z(t)$ que es la transformada de Hilbert de la señal y que tiene componente real ($\text{Re}\{z\}$) e imaginaria ($\text{Im}\{z\}$) por lo que la función $\arg(\cdot)$ utilizada en las Ecuaciones (76) y (77) se calcula así

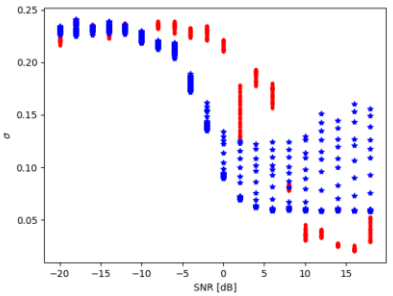
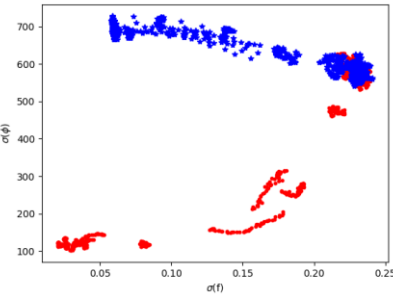
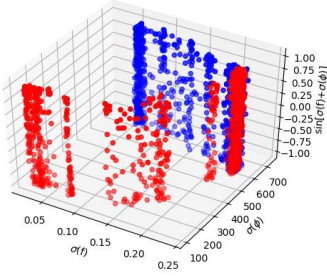
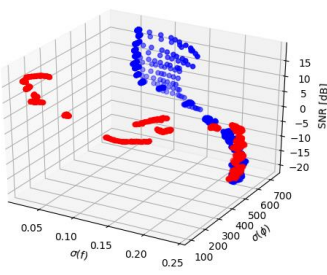
$$\arg(z(t)) = \tan^{-1} \left(\frac{\text{Im}\{z\}}{\text{Re}\{z\}} \right). \quad (78)$$

Ahora bien, una vez se calcula las desviaciones estándar de la amplitud, frecuencia y fase instantánea de la señal, se procede a utilizar esto con diferentes funciones para crear las características con que se entrenarían los algoritmos para luego determinar cuál funciona mejor. La Tabla 4 resume las características utilizadas junto con una imagen que muestra la forma en cómo se ven los datos que tendrían que ser clasificados.

Tabla 4. Conjuntos de características usados para entrenar los algoritmos de clasificación de modulación.

Característica	Ejes	Imagen
Lineal (fase): Desviación estándar de la fase instantánea.	Horizontal: σ_p Vertical: 0	

<p>Sinusoidal (fase): Seno de la desviación estándar de la fase instantánea.</p>	<p>Horizontal: σ_p Vertical: $\sin(\sigma_p)$</p>	
<p>SNR (fase): Desviación estándar de la fase instantánea contra SNR de la señal.</p>	<p>Horizontal: SNR Vertical: σ_p</p>	
<p>Lineal (frecuencia): Desviación estándar de la frecuencia instantánea.</p>	<p>Horizontal: σ_f Vertical: 0</p>	
<p>Sinusoidal (fase): Seno de la desviación estándar de la frecuencia instantánea.</p>	<p>Horizontal: σ_f Vertical: $\sin(\sigma_f)$</p>	

<p>SNR (fase): Desviación estándar de la frecuencia instantánea contra SNR de la señal.</p>	<p>Horizontal: SNR Vertical: σ_f</p>	
<p>Fase y frecuencia: Desviación estándar de la fase instantánea contra desviación estándar de la frecuencia instantánea de la señal.</p>	<p>Horizontal: σ_f Vertical: σ_p</p>	
<p>Seno de la fase y la frecuencia: Seno de la suma de la desviación estándar de la fase instantánea y la desviación estándar de la frecuencia instantánea de la señal.</p>	<p>X: σ_f Y: σ_p Z: $\sin(\sigma_f + \sigma_p)$</p>	
<p>SNR (fase y frecuencia): Desviación estándar de la fase instantánea y la desviación estándar de la frecuencia instantánea de la señal contra los valores de SNR.</p>	<p>X: σ_f Y: σ_p Z: $\sin(\sigma_f + \sigma_p)$</p>	

A pesar de que en las características lineales parece que los datos con clases roja y azul están completamente separados, esto no es cierto, existe un segmento en el que se solapan los datos y esto se puede ver más claramente en la característica sinusoidal.

En las imágenes de la Tabla 4 los datos en rojo corresponden a señales con modulación BPSK y los azules a señales con modulación BFSK.

Cada una de las características anteriormente descritas fueron usadas para entrenar los algoritmos de redes neuronales, máquinas de soporte vectorial y árboles de decisión. Para estos algoritmos se usaron las librerías de *scikit-learn* de Python 2.7 [57]. El motivo por el cual se recurrió a librerías desarrolladas por terceros en lugar de desarrollar los algoritmos es porque estas ya están depuradas, funcionan muy bien y son muy eficientes lo cual es muy importante a aplicaciones de tiempo real y es a eso precisamente que se quiere llegar. Además de que el desarrollo de los algoritmos está fuera del alcance de este proyecto.

Los parámetros utilizados para el entrenamiento de las redes neuronales fueron variados para evaluar el desempeño de cada configuración de los algoritmos para así determinar la mayor precisión posible en la clasificación de la modulación. Para cada característica de los datos, se varió el número de capas desde 1 hasta 3, el número de neuronas en cada capa desde 1 hasta 10, las funciones de activación (Identity, Logistic, tanh, ReLU) y los solver (LBFGS, SGD, Adam).

Para las máquinas de soporte vectorial se varió el parámetro de penalización ‘c’ desde 1 hasta 5 y se evaluaron tres *kernel* (lineal, RBF y Sigmoid). Los demás parámetros se dejaron en su valor por defecto.

Finalmente, para los árboles de decisión se variaron los parámetros de número mínimo de datos en una hoja, desde 1 hasta 10, el número máximo de capas del árbol, desde 1 hasta 20. También se evaluaron los dos criterios para evaluar las creaciones de nuevos subgrupos de datos: Gini y Entropy. Para la forma de crear nuevos subgrupos de datos, se probaron las dos posibilidades del módulo de Scikit: Best y Random.

CAPÍTULO 6

RESULTADOS

6.1 ALGORITMO DE ESTIMACIÓN DE PARÁMETROS

El algoritmo de estimación de parámetros fue probado con señales con valores SNR que van desde -20 dB hasta 18 dB, tal y como se detalla en la sección 5.2. Se encontró que el algoritmo no converge a una solución en todos los casos y sobretodo falla en aquellos en donde el SNR es bajo. La **¡Error! No se encuentra el origen de la referencia.** muestra el porcentaje de los casos en que el algoritmo retorna una solución para señales generadas con modulación BPSK. Sin embargo, se puede ver que a partir de 2 dB de relación señal/ruido es posible obtener aproximaciones con muy buena precisión en todos los parámetros.

Tabla 5. Porcentaje en los que el algoritmo de estimación de parámetros converge a una solución para señales BPSK a determinado SNR y los porcentajes de error absoluto para cada parámetro.

Resultados para BPSK					
SNR [dB]	Porcentaje de aciertos [%]	Porcentaje de doble aproximación [%]	Error porcentual Amplitud [%]	Error porcentual Frecuencia [%]	Error porcentual Ancho de banda [%]
-20	92	60	3,91	81,81	56,09
-18	100	100	3,95	26,13	90,43
-16	52	24	3,71	72,59	47,43
-14	96	80	4,04	35,08	75,65
-12	92	40	4,02	45,63	66,45
-10	100	44	3,72	57,42	48,34
-8	96	56	3,70	65,46	51,75
-6	92	76	5,41	43,66	70,37
-4	92	20	3,31	77,81	33,02
-2	100	40	3,11	51,71	44,17
0	72	32	1,22	14,49	35,54

2	100	0	0,91	9,18	0,52
4	100	0	0,16	7,20	0,39
6	100	0	0,40	5,69	1,24
8	100	0	0,30	8,12	0,43
10	100	0	1,27	5,01	0,77
12	100	0	0,21	7,08	0,33
14	100	0	0,21	6,07	0,33
16	100	0	2,09	10,79	0,90
18	100	0	1,52	2,38	0,43

Para las señales con modulación BFSK se obtuvieron mayor número de aciertos ya que estas presentan más información en frecuencia y por lo tanto los picos son más pronunciados. Los resultados obtenidos para señales con esta modulación son presentados en la Tabla 6.

Tabla 6 Porcentaje en los que el algoritmo de estimación de parámetros converge a una solución para señales BFSK a determinado SNR y los porcentajes de error absoluto para cada parámetro.

Resultados para BFSK					
SNR [dB]	Porcentaje de aciertos [%]	Porcentaje de doble aproximación [%]	Error porcentual Amplitud [%]	Error porcentual Frecuencia [%]	Error porcentual Ancho de banda [%]
-20	88	28	3,86	79,79	39,18
-18	80	48	4,41	87,08	30,14
-16	40	40	5,98	38,26	4,75
-14	72	16	6,78	97,53	51,74
-12	80	48	3,94	46,28	31,35
-10	92	72	4,33	43,29	17,79
-8	96	88	4,73	12,67	7,31
-6	100	100	6,40	1,03	0,43
-4	100	100	8,97	1,04	0,65
-2	100	100	11,39	2,46	0,75
0	100	100	12,01	9,03	1,43
2	100	100	11,63	11,37	2,09
4	100	100	11,44	0,88	0,88
6	100	72	12,44	13,83	13,98
8	96	92	12,05	2,09	0,02
10	100	100	11,61	3,83	1,18
12	100	100	12,85	10,61	3,50
14	100	100	11,85	0,95	0,64
16	100	96	11,98	2,26	2,15
18	100	80	12,56	7,06	10,00

El algoritmo también fue probado usando señales reales adquiridas por un SDR en las tres bandas VHF, UHF y Banda S. El resumen de los resultados obtenidos para estos casos se resume en la Tabla 7.

Tabla 7. Resultados del algoritmo de estimación de parámetros para datos adquiridos por el SDR.

Banda	Porcentaje de aciertos [%]	Porcentaje de acierto en modulación [%]	Error porcentual Amplitud [%]	Error porcentual Frecuencia [%]	Error porcentual Ancho de banda [%]
VHF	87.50	100	47.65	0,66	49,99
UHF	97.50	100	0,60	0,73	23,92
S	99.50	100	15,00	0,33	6,52

Como una muestra de las aproximaciones realizadas por el algoritmo, se presentan la Figura 30, y la Figura 31 las cuales corresponden a resultados obtenidos para señales BPSK generadas con SNR de 4 dB y 18 dB respectivamente.

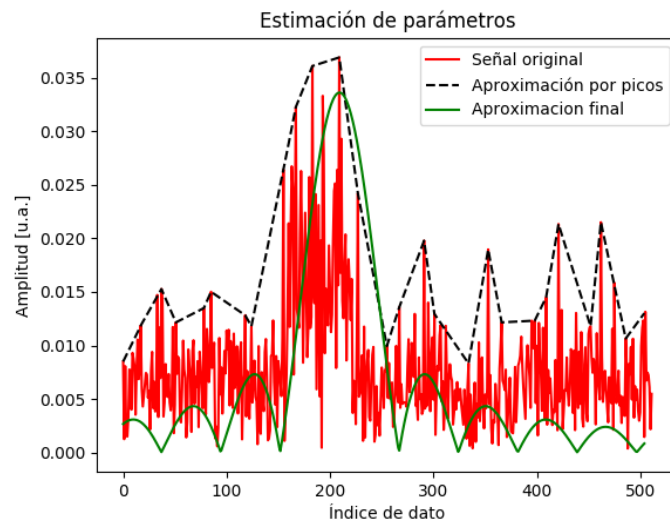


Figura 30. Resultado del algoritmo de estimación de parámetros para una señal BPSK con SNR de 4 dB.

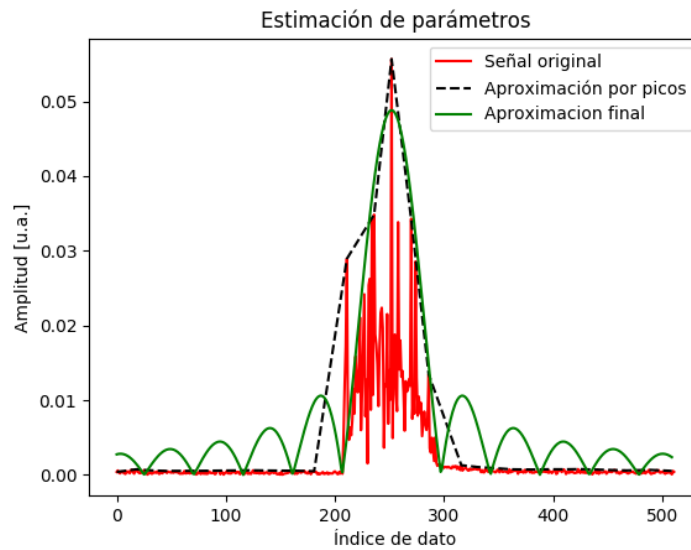


Figura 31. Resultado del algoritmo de estimación de parámetros para una señal BPSK con SNR de 18 dB.

En la Figura 32, la Figura 33 y la Figura 34 se muestran resultados para señales BFSK generadas con SNR de -6 dB, 10 dB y 18 dB respectivamente.

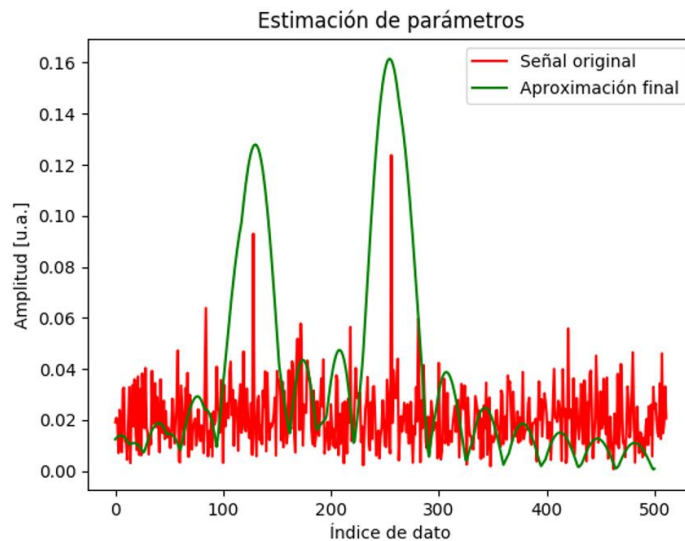


Figura 32. Resultado del algoritmo de estimación de parámetros para una señal BFSK con SNR de -6 dB.

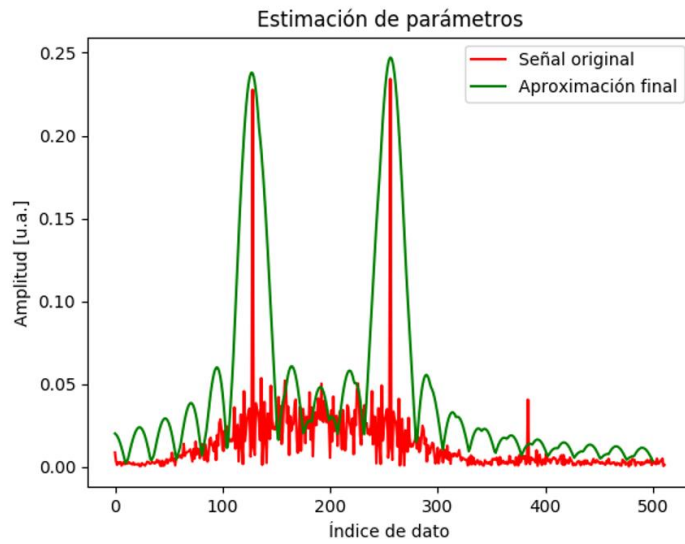


Figura 33. Resultado del algoritmo de estimación de parámetros para una señal BPSK con SNR de 10 dB.

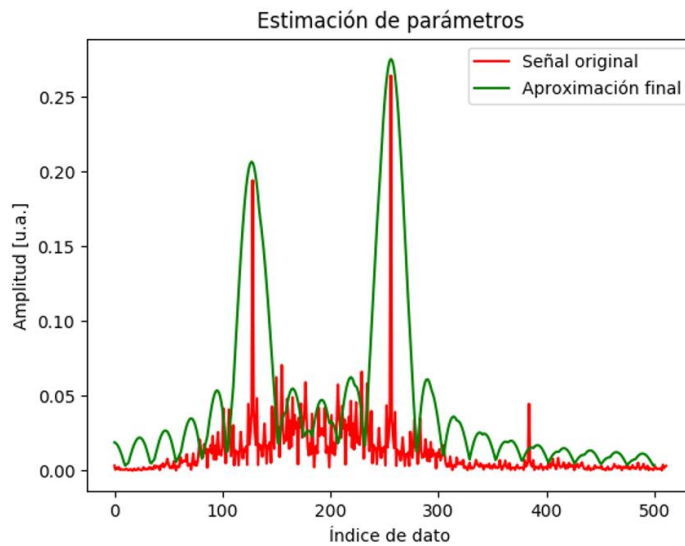


Figura 34. Resultado del algoritmo de estimación de parámetros para una señal BPSK con SNR de 18 dB.

Este algoritmo también fue probado con algunas señales reales capturadas con un SDR y se obtuvieron resultados similares a los ya expuestos. En la Figura 35 se muestra uno de estos casos.

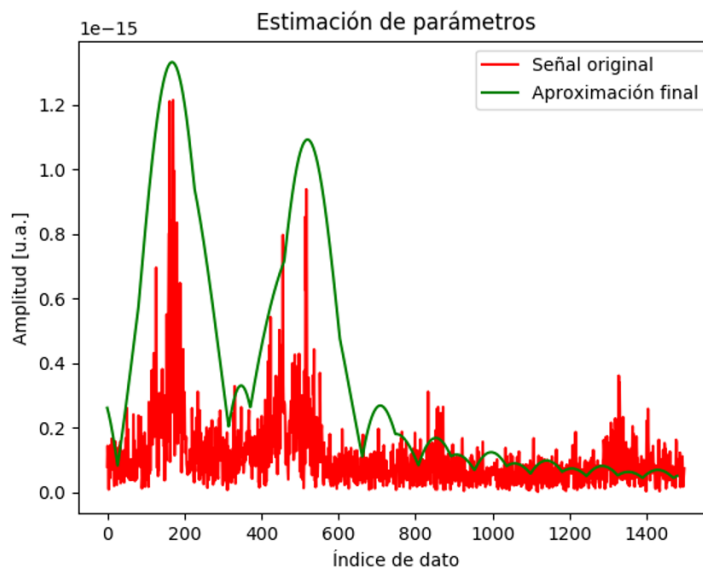


Figura 35. Resultado del algoritmo de estimación de parámetros para una señal con modulación BFSK adquirida con un SDR.

6.2 CLASIFICACIÓN DE MODULACIÓN

Al entrenar los algoritmos de redes neuronales, máquinas de soporte vectorial y árboles de decisión con cada una de las características presentadas en la sección de metodología, se imprimieron los resultados de precisión y las matrices con número de dimensiones igual al número de parámetros variados en cada caso. Luego de obtener las matrices se evaluó cuál fue la precisión máxima en las predicciones y la configuración de parámetros en los algoritmos que conducen a la máxima precisión. La Tabla 8, Tabla 9 y la Tabla 10 muestran los mejores resultados obtenidos para cada algoritmo.

Tabla 8. Mejor resultado para el algoritmo de Redes Neuronales.

Redes Neuronales	Precisión máxima: 0,92	Característica: Desviación estándar de la fase y la frecuencia instantáneas	1 capa oculta	8 neuronas	Función de activación: Logistic	Solver: LBFGS
-------------------------	----------------------------------	---	---------------	------------	---	-------------------------

Tabla 9. Mejor resultado para el algoritmo de Árboles de Decisión.

Árboles de decisión	Precisión máxima: 0,90	Característica: Desviación estándar de la fase instantánea	4 datos mínimo	Profundidad máxima: 4	Splitter: Random	Criterio: Gini
----------------------------	----------------------------------	--	----------------	------------------------------	----------------------------	--------------------------

Tabla 10. Mejor resultado para el algoritmo de Máquinas de Soporte Vectorial.

Máquinas de soporte vectorial	Precisión máxima: 0,89	Característica: Desviación estándar de la fase instantánea	C=3	Kernel: RBF
--------------------------------------	----------------------------------	--	-----	-----------------------

Se pudo ver que cada característica reporta valores de precisión muy diferentes, sujeto también a los diferentes parámetros de los algoritmos.

Cada una de las características usadas para el entrenamiento contenía 1000 datos de los cuales el 70% fue realmente usado en la etapa de entrenamiento y el 30% se usó como set de prueba para imprimir los resultados de precisión que fueron guardados en las matrices de resultados de los algoritmos.

Ahora, en cuanto a la característica que mejor desempeño tiene, se pudo ver que, para este caso, una de las características más fáciles de calcular, como lo es la representación lineal de la desviación estándar de la fase instantánea, dio unos resultados mejores o comparables con aquellas características que contenían mayor información de las señales. De igual forma se encontró que el algoritmo de redes de neuronales que presenta el mejor desempeño, contiene características que no requieren de grandes capacidades de cómputo, lo cual representa una ventaja y justifica el hecho de que se evalúen diferentes parámetros en los algoritmos.

Finalmente, los algoritmos desarrollados se publicaron GitHub con el fin de hacerlos públicos de que otras personas puedan acceder a ellos para usarlos o modificarlos.

CAPÍTULO 7

CONCLUSIONES

- En este trabajo se concluye que el algoritmo más adecuado para clasificar entre las modulaciones BPSK y BFSK es el de redes neuronales bajo una característica de los datos en donde se tiene en consideración de la desviación estándar de la fase y la frecuencia instantánea de las señales.
- El algoritmo de estimación de parámetros desarrollado funciona bien para señales con SNR iguales o superiores a 10 dB por lo que no es ideal para usarse en aplicaciones en donde las fuentes son muy lejanas.
- Se pudo ver que para las señales con modulación BFSK es un poco más fácil determinar sus parámetros por medio de aproximaciones, dado que, al tener mayor contribución en frecuencia, el ruido no afecta los picos centrales tan rápidamente. Caso contrario a lo que ocurre con las señales con modulación BPSK en donde el ruido esconde rápidamente la forma principal de su transformada de Fourier.
- Si bien en este trabajo se usó la fase instantánea de las señales evaluadas como un parámetro para la clasificación de modulaciones, esto puede no ser tan efectivo con otro tipo de modulaciones. Por ejemplo, para clasificar señales FSK con diferentes tasas de bits por símbolo, tales como 4-FSK, 8-FSK y 16-FSK, esta característica mostraría valores muy semejantes por lo que se dificultaría la clasificación de los datos.
- El uso de diferentes características en la clasificación de clases para algoritmos de *machine learning* es esencial puesto que permite distribuir mejor los datos en el espacio y, de esa manera, tal vez mejorar el desempeño de los algoritmos y así mismo su precisión.
- A pesar de que los algoritmos de árboles de decisión y máquinas de soporte vectorial fueron entrenados con características que no requieren de mucho procesamiento, tales como el hecho de que la profundidad máxima del árbol fuera de tan solo 3 capas y que en las máquinas de soporte vectorial se usó un *kernel* lineal; mostraron resultados muy similares a los alcanzados por el algoritmo de redes neuronales. Éste, en cambio, sí tuvo parámetros más exigentes como las 3 capas ocultas con 10 neuronas. Esto se puede deber al hecho de que, si bien los datos se solapaban mucho, no existía una mejor solución para separarlos que una línea recta o una aproximación a una línea recta.

- El modelo usado de valor absoluto de *sinc* para el algoritmo de estimación de parámetros, fue suficiente para aproximar señales que fueron creadas utilizando un filtro de forma de raíz de coseno, el cual es el caso de los sets de datos creados y descritos en la sección 5.2.
- En este trabajo se pudo experimentar que, si bien es importante utilizar señales reales para probar los algoritmos desarrollados, también es de gran utilidad e importancia simular los datos y las señales desde el computador ya que de esa manera se pueden simular condiciones muy diversas tales como SNR exactos, pérdidas de espacio libre, corrimientos en la frecuencia de muestreo y frecuencia central, entre otras características que, con señales reales tomaría mucho tiempo y requeriría de un gran proceso de experimentación para obtenerlas.
- El uso de herramientas desarrolladas por terceros, tales como las librerías de *scikit* y los diferentes módulos del software GNURadio fueron esenciales en el desarrollo de este proyecto y se pudo evidenciar que la integración de herramientas contribuye en gran medida a la implementación de este tipo de proyectos en donde, al hacer esto, es posible enfocarse en las características propias del problema.
- Las gráficas arrojadas después de la caracterización de la potencia de recepción del radio SDR muestran la importancia de realizar este procedimiento, ya que no es algo que se pueda determinar analíticamente o con datos del fabricante puesto que depende de las características de la implementación.

REFERENCIAS

- [1] H. D. Muñoz, *Principios básicos de comunicaciones digitales*. Medellín: Universidad Pontificia Bolivariana, 1996.
- [2] F. Xiong, *Digital Modulation Techniques Second Edition*. House Boston. london Artechhouse, Artech.
- [3] Agilent Techonologies, “Digital Modulation in Communications Systems — An Introduction.”
- [4] F. H. Ali and P. Denbigh, “Phase Shift Keying,” in *Wiley Encyclopedia of Electrical and Electronics Engineering*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 1999.
- [5] K. P. Murphy, *Machine learning : a probabilistic perspective*. MIT Press, 2012.
- [6] D. M. Skapura, *Building neural networks*. ACM Press, 1996.
- [7] M. Anthony and P. L. Bartlett, *Neural network learning : theoretical foundations*. Cambridge University Press, 1999.
- [8] L. Wang, Y. Yang, R. Min, and S. Chakradhar, “Accelerating deep neural network training with inconsistent stochastic gradient descent,” *Neural Networks*, vol. 93, pp. 219–229, Sep. 2017.
- [9] T. Salimans and D. P. Kingma, “Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks,” in *Neural Information Processing Systems (NIPS Proceedings)*, 2016, pp. 901–909.
- [10] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, “Support Vector Clustering,” *J. Mach. Learn. Res.*, vol. 2, no. Dec, pp. 125–137, 2001.
- [11] Harrison Kinsley, “Python Programming Tutorials.” [Online]. Available: <https://pythonprogramming.net/support-vector-machine-intro-machine-learning-tutorial/>. [Accessed: 13-Oct-2017].
- [12] A. Ng, “Support Vector Machines,” in *CD229 Lecture Notes*, Stanford University.
- [13] J. A. K. Suykens and J. Vandewalle, “Least Squares Support Vector Machine Classifiers,” *Neural Process. Lett.*, vol. 9, pp. 293–300, 1999.
- [14] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, “Fisher discriminant analysis with kernels,” in *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468)*, pp. 41–48.
- [15] S. Tong and D. Koller, “Support Vector Machine Active Learning with Applications to Text Classification,” *J. Mach. Learn. Res.*, vol. 2, no. Nov, pp. 45–66, 2001.
- [16] Y.-J. Lee and O. L. Mangasarian, “SSVM: A Smooth Support Vector Machine for Classification,” *Comput. Optim. Appl.*, vol. 20, no. 1, pp. 5–22, 2001.

- [17] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text Classification using String Kernels," *J. Mach. Learn. Res.*, vol. 2, no. Feb, pp. 419–444, 2002.
- [18] J.-P. Vert, K. Tsuda, and B. Schölkopf, *A primer on kernel methods*. 2004.
- [19] O. M. Lior Rokach, *DATA MINING WITH DECISION TREES*. .
- [20] S. J. (Stuart J. Russell, P. Norvig, and E. Davis, *Artificial intelligence : a modern approach*. Prentice Hall, 2010.
- [21] H. Zhao and X. Li, "A cost sensitive decision tree algorithm based on weighted class distribution with batch deleting attribute mechanism," *Inf. Sci. (Ny)*, vol. 378, pp. 303–316, Feb. 2017.
- [22] J. J. Oliver, J. J. Oliver, and D. J. Hand, "On Pruning and Averaging Decision Trees," *Proc. TWELFTH Int. Conf. Mach. Learn.*, pp. 430--437, 1995.
- [23] Y.-Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction.," *Shanghai Arch. psychiatry*, vol. 27, no. 2, pp. 130–5, Apr. 2015.
- [24] R. T. Jacobs, J. B. Coder, and V. M. Musser, "Spectrum sensing with WLAN access points," in *2016 United States National Committee of URSI National Radio Science Meeting (USNC-URSI NRSM)*, 2016, pp. 1–2.
- [25] H. Sun, A. Nallanathan, S. Cui, and C.-X. Wang, "Cooperative Wideband Spectrum Sensing Over Fading Channels," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1382–1394, Mar. 2016.
- [26] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, 1999.
- [27] C. M. Watson, "Signal Detection and Digital Modulation Classification-Based Spectrum Sensing for Cognitive Radio," 2013.
- [28] M. A. Abdulsattar and Z. A. Hussein, "Energy Detection Technique for Spectrum Sensing in Cognitive Radio: A Survey," *Int. J. Comput. Networks Commun.*, vol. 4, no. 5, 2012.
- [29] D. M. Martínez-Plata and Á. G. Andrade-Reátiga, "Evaluation of energy detection for spectrum sensing based on the dynamic selection of detection-threshold," *Procedia Eng.*, vol. 35, pp. 135–143, Jan. 2012.
- [30] W.-L. Chin, J.-M. Li, and H.-H. Chen, "Low-Complexity Energy Detection for Spectrum Sensing With Random Arrivals of Primary Users," *IEEE Trans. Veh. Technol.*, vol. 65, no. 2, pp. 947–952, Feb. 2016.
- [31] A. Kumar, R. Goyal, and D. Ray, "Spectrum Sensing using Energy Detection Algorithm for Cognitive Radio," *Int. Res. J. Eng. Technol.*, pp. 2395–56, 2017.
- [32] S. Ziafat, W. Ejaz, and H. Jamal, "Spectrum sensing techniques for cognitive radio networks: Performance analysis," in *2011 IEEE MTT-S International Microwave Workshop Series on Intelligent Radio for Future Personal Terminals*, 2011, pp. 1–4.

- [33] A. Mishra and A. Mishra, "PSD based primary user detection in Cognitive Radio systems operating in impulsive noise environment," *Int. Res. J. Eng. Technol.*
- [34] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1, pp. 772–776.
- [35] H. Reyes, S. Subramaniam, N. Kaabouch, and W. C. Hu, "A spectrum sensing technique based on autocorrelation and Euclidean distance and its comparison with energy detection for cognitive radio networks," *Comput. Electr. Eng.*, vol. 52, pp. 319–327, May 2016.
- [36] R. Chen, J. Wang, R. Lin, and X. Zhao, "Spectrum Sensing Based on Nonparametric Autocorrelation in Wireless Communication Systems under Alpha Stable Noise," *Mob. Inf. Syst.*, vol. 2016, pp. 1–6, May 2016.
- [37] S. Sedighi, A. Taherpour, S. Gazor, and T. Khattab, "Eigenvalue-Based Multiple Antenna Spectrum Sensing: Higher Order Moments," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 2, pp. 1168–1184, Feb. 2017.
- [38] M. Hamid, N. Bjorsell, and S. Ben Slimane, "Energy and Eigenvalue Based Combined Fully Blind Self Adapted Spectrum Sensing Algorithm," *IEEE Trans. Veh. Technol.*, vol. 65, no. 2, pp. 630–642, Feb. 2016.
- [39] E. E. Azzouz and A. K. Nandi, "Procedure for automatic recognition of analogue and digital modulations," *IEE Proc. - Commun.*, vol. 143, no. 5, p. 259, 1996.
- [40] C. Park and D. Kim, "A novel robust feature of modulation classification for reconfigurable software radio," *IEEE Trans. Consum. Electron.*, vol. 52, no. 4, pp. 1193–1200, Nov. 2006.
- [41] A. K. Nandi and E. E. Azzouz, "Modulation recognition using artificial neural networks," *Signal Processing*, vol. 56, no. 2, pp. 165–175, Jan. 1997.
- [42] B. Zhu, S. Peng, Y. Su, and T. Tang, "Automatic Identification Method of Link4A," in *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, 2009, pp. 239–243.
- [43] V. Chaithanya and V. U. Reddy, "Blind modulation classification in the presence of carrier frequency offset," in *2010 International Conference on Signal Processing and Communications (SPCOM)*, 2010, pp. 1–5.
- [44] L.-X. Wang and Y.-J. Ren, "Recognition of digital modulation signals based on high order cumulants and support vector machines," in *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, 2009, pp. 271–274.
- [45] Ataollah Ebrahim Zadeh, "Automatic recognition of radio signals using a hybrid intelligent technique," *Expert Syst. Appl.*, vol. 37, no. 8, pp. 5803–5812, Aug. 2010.
- [46] Bijan G. Mobasser, "Digital modulation classification using constellation shape," *Signal Processing*, vol. 80, no. 2, pp. 251–277, Feb. 2000.

- [47] C. Yin, B. Li, and Y. Li, "Modulation Classification of MQAM Signals from Their Constellation Using Clustering," in *2010 Second International Conference on Communication Software and Networks*, 2010, pp. 303–306.
- [48] N. Liu, B. Liu, S. Guo, and R. Luo, "Investigation on Signal Modulation Recognition in the Low SNR," in *2010 International Conference on Measuring Technology and Mechatronics Automation*, 2010, pp. 528–531.
- [49] T. S. Rappaport, *Wireless communications: principles and practice*. Dorling Kindersley, 2009.
- [50] K. Y. Jo, *Satellite communications network design and analysis*. Artech House, 2011.
- [51] Dassault Systeme, "SolidWorks." 2015.
- [52] Ettus Research, "n200/n210 pRодукt oveRview."
- [53] Ettus Research, "UBX Daughterboard."
- [54] Minicircuits, "Power Detector ZX47-60LN-S+."
- [55] T. J. O'Shea and N. West, "Radio Machine Learning Dataset Generation with GNU Radio," *Proc. GNU Radio Conf.*, vol. 1, no. 1, Sep. 2016.
- [56] A. Hazza, M. Shoaib, S. A. Alshebeili, and A. Fahad, "An overview of feature-based methods for digital modulation classification," in *2013 1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, 2013, pp. 1–6.
- [57] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. Oct, pp. 2825–2830, 2011.
- [58] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung, "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit," *Nature*, vol. 405, no. 6789, pp. 947–951, Jun. 2000.